



TAMPEREEN
AMMATTIKORKEAKOULU

DC-SERVOMOOTTORIN TESTAUS- JA OPE- TUSLAITEEN SUUNNITTELU

Jari Kuosmanen

Opinnäytetyö
Toukokuu 2017
Kone- ja tuotantotekniikka
Koneautomaatio



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Kone- ja tuotantotekniikka
Koneautomaatio

KUOSMANEN, JARI:
DC-servomoottorin testaus- ja opetuslaite

Opinnäytetyö 50 sivua, joista liitteitä 11 sivua
Toukokuu 2017

Opinnäytetyön tavoitteena oli suunnitella laitteisto DC-servomoottorin opetuskäyttöön Tampereen ammattikorkeakoulun koneautomaation laboratorioon. Säätötekniikan opetuksen kannalta on oleellista pystyä testaamaan tietokoneella laskettujen säätöparametrien toimintaa oikealla laitteistolla. Laitteistoja on tarkoitus rakentaa jatkossa useampia kappaleita, jonka vuoksi suunnittelussa piti ottaa huomioon tiukka budjetti sekä modulaarisuus, eli mahdollisuus helposti vaihtaa komponentteja laitteeseen.

Tarkkojen komponenttivalintojen, mekaanisen suunnittelun sekä sähkösuunnittelun myötä lopputuloksena on suunnitelma, joka täyttää laitteelle asetetut vaatimukset. Laitteen suunnittelussa onnistui hyvin modulaarisuuden saavuttaminen sekä budjetissa pysyminen. Laitteita on helppo rakentaa jatkossa lisää tarpeen mukaan.

Tulevaisuudessa laitetta voi modulaarisuuden ansiosta muokata sopimaan erilaisiin opetus- ja testaustarkoituksiin. Laitteen ohjaustapaa voi helposti vaihtaa esimerkiksi halvasta mikrokontrollerista teollisuustason logiikkakontrolleriin, mikä mahdollistaa monipuolisen käytön jatkossakin.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Mechanical Engineering
Machine Automation

KUOSMANEN, JARI:
A Device for Testing and Teaching DC servomotor

Bachelor's thesis 50 pages, appendices 11 pages
May 2017

The goal of this thesis was to design a device which can be used to teach DC motor control techniques at the machine automation laboratory in Tampere University of Applied Sciences. For the purposes of teaching control techniques, it is important to test computer-calculated parameters with real-life machinery. There are plans to build several copies of the device in the future, and budget limitations created a need for modularity in the design, i.e., the possibility to easily change components of the device.

By successful selection of components and mechanical and electrical design, a device which fulfills the original requirements was achieved. The goal of achieving modularity was also reached, and the cost of the device stayed under budget. It will be easy to build more devices in the future if needed.

Because of modularity it is easy in the future to modify the device to fit different kinds of teaching and testing purposes. The control method of the device can be changed easily, for example from a cheap microcontroller to an industrial logic controller, allowing versatile uses in the future.

Key words: servo, dc motor, control techniques, Arduino, microcontroller

SISÄLLYS

1	JOHDANTO.....	6
2	SERVOJÄRJESTELMÄT	8
2.1	Servotekniikka	8
2.2	Avoin ohjaus ja suljettu säätöpiiri	9
2.3	Säätösuunnittelu.....	11
2.4	Sähköservojärjestelmän komponentit	12
2.4.1	Anturit	13
2.4.2	Servomoottori.....	13
2.4.3	Ohjausjärjestelmät.....	17
3	LAITTEEN SUUNNITTELU	19
3.1	Laitteen vaatimukset	19
3.2	Komponenttien valinta.....	19
3.2.1	Moottori ja liikeruuvi	20
3.2.2	Vastamomentti	22
3.2.3	Johteet, laakerit ja kytkimet	22
3.2.4	Moottorin kontrolleri.....	24
3.3	Mekaniikkasuunnittelu.....	25
3.3.1	Runko	25
3.3.2	Jarruna toimivan askelmoottorin teline.....	26
3.3.3	Kelkka	27
3.3.4	Korokepalat	27
3.4	Sähkösuunnittelu.....	28
3.4.1	L298N-moduuli ja moottori	28
3.4.2	Arduino	29
3.4.3	Rajakytkimet ja sulake	30
3.4.4	Jarrumoottorin kytkentä	31
4	TESTIOHJELMA	32
4.1	Kotiasemaan ajo.....	32
4.2	Enkooderin lukeminen	32
4.3	Moottorin ajo	33
5	YHTEENVETO JA POHDINTA	35
	LÄHTEET.....	36
	LIITTEET	37
	Liite 1. Moottorin datalehti	37
	Liite 2. Askelmoottorin datalehti.....	38

Liite 3. Kokoonpanokuva, jossa osat erikseen merkattuina	39
Liite 4. Runkolevyn mitoitettu 2D-malli	40
Liite 5. Jarrumoottoritelineen mitoitettu 2D-malli	41
Liite 6. Kelkan yläosan mitoitettu 2D-malli.....	42
Liite 7. Kelkan liitântäosan mitoitettu 2D-malli.....	43
Liite 8. Sähkökaavio.....	44
Liite 9. Testiohjelma.....	45

1 JOHDANTO

Servotekniikka on tärkeä osa teollisuuden laitteistoja, sen avulla saavutetaan erittäin hyviä tarkkuuksia, nopeutta sekä stabiiliutta. Servolaitteita voivat olla sähköiset, pneumaattiset sekä hydrauliset toimilaitteet, joiden käyttäytymistä pystytään säätämään takaisinkytkennän avulla. Servolaitteilla voidaan saavuttaa esimerkiksi tietty nopeus, matka tai voima. Servot ovat esimerkiksi robotiikassa avainasemassa, sillä robotin käsivarren tarkka paikka, liikenopeus sekä voima ovat modernin robotiikan tärkeimmät asiat. Servojärjestelmiä hyödynnetään teollisuudessa myös CNC-koneistuslaitteissa sekä esimerkiksi kutomakoneissa, joissa kudontanopeus riippuu täysin useiden servolaitteiden yhdessä toimimisesta. Servotekniikan olennainen osa on säätötekniikka, jonka avulla toimilaitteet saadaan käyttäytymään halutunlaisesti sekä reagoimaan muutoksiin.

Koneautomaation opetuksessa servo- ja säätötekniikka on merkittävässä asemassa. Opetuksen kannalta on tärkeää voida matemaattisen mallinnuksen ja simuloinnin lisäksi testata ohjausta myös varsinaisella servolaitteella, koska mallinnuksessa ja simuloinnissa on vaikeaa ottaa huomioon kaikkia mahdollisia toimintaan vaikuttavia tekijöitä. Servolaitteen lopulliset säädöt pystytään tekemään vasta käytännön testauksen myötä. Tampereen ammattikorkeakoulun säätö- ja servotekniikan opetuksessa on keskitytty mallinnukseen ja simulointiin, koska sopivia laitteita käytännön testaukseen ei ole ollut käytössä.

Opinnäytetyön tavoitteena oli suunnitella laite, jota voidaan hyödyntää servotekniikan opetuksessa. Laitteella tulee pystyä liikuttamaan tasavirtamoottorin avulla massaa lineaarisesti vaaka- sekä pystysuunnassa. Laitteita tarvitaan opetuskäyttöön useita kappaleita, jonka vuoksi laitteesta pitää pyrkiä suunnittelemaan tarpeeksi pieni varastoinnin helpottamiseksi. Laitteen avulla opiskelijoiden tulee pystyä havaitsemaan miten säätöarvojen muutokset vaikuttavat laitteen toimintaan.

Tampereen ammattikorkeakoulussa servotekniikan opetuksessa käytetään Matlab-ohjelmistoa ja siihen kuuluvaa Simulink-mallinnusympäristöä, joten suunnittelussa piti ottaa huomioon miten laitetta pystytään ohjaamaan näitä ohjelmia hyödyntämällä. Koska servotekniikassa tärkeää on myös järjestelmän reagointi muutoksiin ja ulkoisiin häiriöihin,

pitää laitteessa olla myös ominaisuus, jonka avulla liikettä voidaan vastustaa erikseen päälle kytkettävällä vastamomentilla.

Tavoitteena oli myös modulaarinen rakenne, jonka myötä laitteen muuttaminen tulevaisuudessa olisi helpompaa. Työhön sisältyi komponenttien valinta, mekaniikkasuunnittelu, sähkösuunnittelu sekä testiohjelman kirjoittaminen.

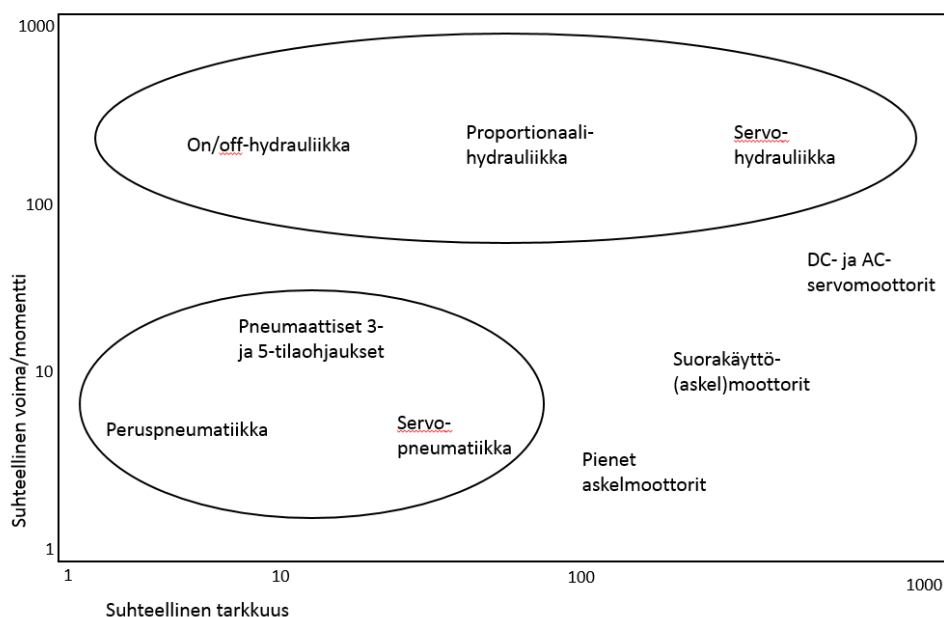
2 SERVOJÄRJESTELMÄT

2.1 Servotekniikka

Servotekniikassa järjestelmän tehtävänä on ohjata haluttuun arvoon esim. kappaleen asema ja nopeus. Toimilaitteina voivat olla erilaiset moottorit tai sylinterit. Säädetävän suureen mukaan puhutaan

- asemaservosta,
- nopeusservosta,
- voimaservosta tai
- momenttiservosta. (Fonselius, Rinkinen & Vilenius 1998, 7.)

Servojärjestelmän toimilaitteet voivat olla sähköisiä, pneumaattisia tai hydraulisia. Vaikeimpia servototeuksia ovat pneumaattiset servot, koska ilman kokoonpuristumisen vuoksi ne ovat toiminnaltaan epälineaarisia. Hydrauliservoina käytetään yleisimmin lineaarisylintereitä. Hydraulisylintereillä saadaan helposti aikaan suuria voimia ja niiden ohjaus on suhteellisen helppoa moderneilla proportionaaliventtiileillä. Yleisin servojärjestelmän toimilaite on sähkömoottori. Kuvassa 1 on vertailtu voiman ja tarkkuuden mukaan eri tekniikoita.

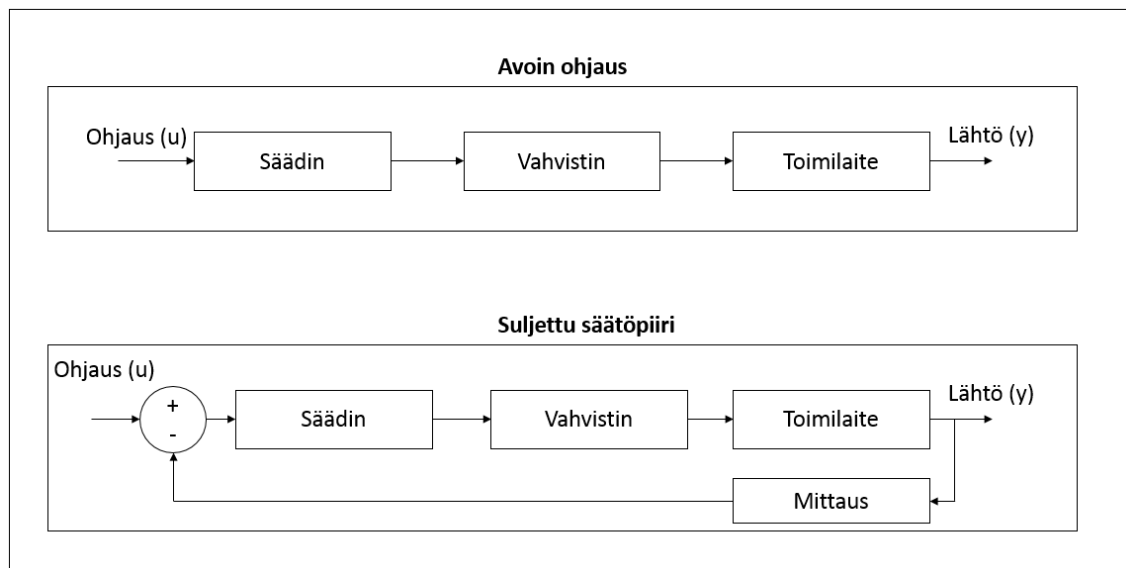


KUVA 1 Toimilaitteiden vertailua

Koneautomaatiossa yleisimpiä ovat asema- ja nopeusservot. Oleellista servojärjestelmässä on takaisinkytkentä. Sen avulla mitataan lähtösuure, jota verrataan järjestelmän ohjaukseen. (Fonselius ym. 1998, 7.)

2.2 Avoin ohjaus ja suljettu säätöpiiri

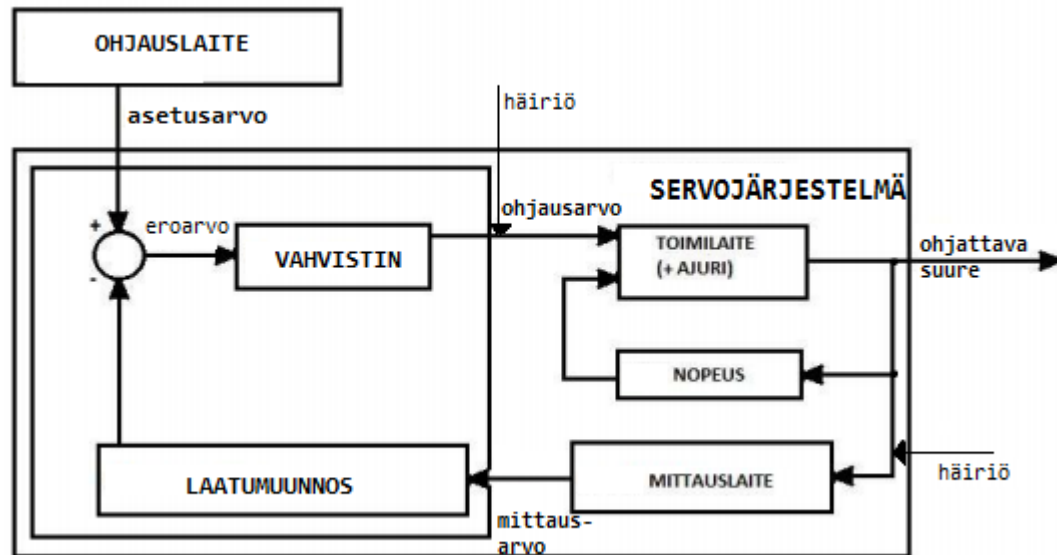
Ilman takaisinkytkentää toteutettua ohjausta kutsutaan avoimeksi ohjaukseksi. Avoin ohjauksessa määritellään ennakkoon, miten toimilaitetta ajetaan, jotta toiminta saavutetaan. Avointa ohjausta voidaan käyttää esimerkiksi säiliön täyttämiseksi pumpun avulla, pumpaten nestettä tietty aikamäärä joka on laskettu pumpun tuoton perusteella. Avoimen ohjauksen heikkoutena on sen, se ei pysty reagoimaan järjestelmään tulleisiin muutoksiin, vaan toimii, kuten se on ennalta asetettu toimimaan. Takaisinkytketyssä järjestelmässä ohjauslaitteelta tulee asetusarvo, johon servojärjestelmä pyrkii jatkuvasti tarkkaillen omaa tilaansa. Kuvassa 2 on esitettyä avoimen ohjauksen ja suljetun säätöpiirin ero.



KUVA 2 Avoin ohjaus ja suljettu säätöpiiri

Kuvassa 3 on esitettyä takaisinkytketyn servojärjestelmän tarkempi periaatteellinen rakenne. Ohjauslaite syöttää järjestelmään asetusarvon, johon servon halutaan asettuvan. Eroelin vertaa asetusarvoa ja servon todellista tilaa ja syöttää eroarvon vahvistimelle. Vahvistimelta menee toimilaitteelle ohjausarvo, joka määrittää toimilaitteen ajamisen. Toimilaitteen tilaa tarkkaillaan mittauslaitteella, josta saadaan eroelimen hyödyntämä

mittaussuure. Kun erosuure on nolla, laite on halutussa tilassa. Servojärjestelmää häiritsevänä tekijänä esiintyy myös erilaisia häiriösignaaleja, joita voidaan etenkin anturipuolella vähentää erilaisilla kompensattoreilla.



KUVA 3 Servojärjestelmän periaatteellinen rakenne.

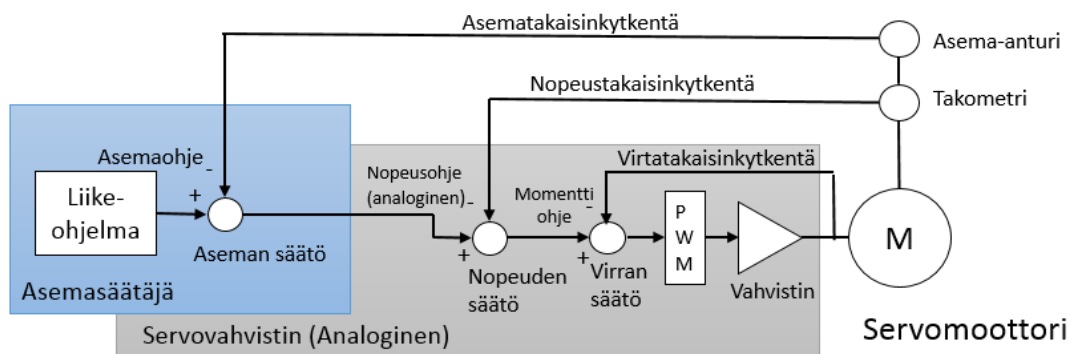
Servojärjestelmän säätimellä on erittäin tärkeä rooli järjestelmässä. Hyvin toteutetulla säädöllä saadaan nopea ja stabiili järjestelmä, huonosti toteutetulla säädöllä järjestelmästä voi tulla hidas ja epästabiili.

Hitaiden järjestelmien vahvistin voi olla hyvin yksinkertainen (P-säätö). Nopeissa järjestelmissä voidaan käyttää PI- tai PID-säätimiä ja useita takaisinkytkentöjä (esim. nopeus- ja kiihtyvyydestakaisinkytkennät) haluttujen ominaisuuksien saavuttamiseksi. (Fonselius ym. 1998, 8.)

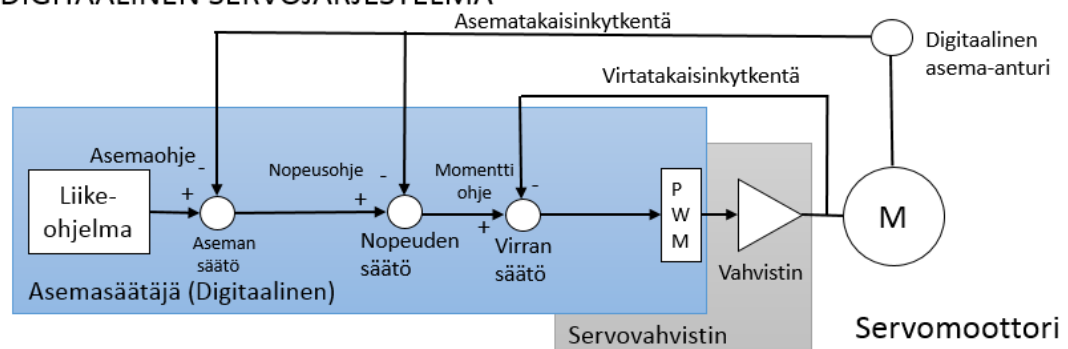
Säätöjärjestelmä pyrkii pitämään erosignaalin nollana, jolloin järjestelmä automaattisesti korjaa häiriöiden aiheuttamat pyörimisnopeuden muutokset. Mitä tarkempi säätö halutaan, sitä enemmän täytyy erosignaalia vahvistaa. Vahvistusta ei voida kuitenkaan lisätä määrättömästi, koska liian suuri vahvistus aikaansaa järjestelmässä värähtelyä tai jopa epästabiiliutta. Usein joudutaan säätöjärjestelmässä tekemään kompromisseja tarkkuus- ja stabiiliusvaatimusten välillä. (Fonselius ym. 1998, 11.)

Servojärjestelmiä voi olla analogisia sekä digitaalisia. Kuvassa 4 on molempien järjestelmien kaaviot, joista näkee niiden toiminnan eron. Analogiajärjestelmässä asemasäätäjältä tulee servovahvistimelle tietona vain asemaohjeen ja takaisinkytkentätiedon erotus, jonka perusteella servovahvistimen analogiset komponentit muuttavat moottorille syötettävää jännitettä. Analogisessa järjestelmässä nopeustietoa ja virtaa käsitellään suoraan servovahvistimen analogisilla komponenteilla. Digitaalisessa järjestelmässä kaikki takaisinkytkentätieto käsitellään asemasäätäjällä, joka syöttää servovahvistimelle vain tarvittavan jännitemäärän. Digitaalisen järjestelmän etuna on sen nopeus ja helpompi säätösuunnittelu.

ANALOGINEN SERVOJÄRJESTELMÄ



DIGITAALINEN SERVOJÄRJESTELMÄ

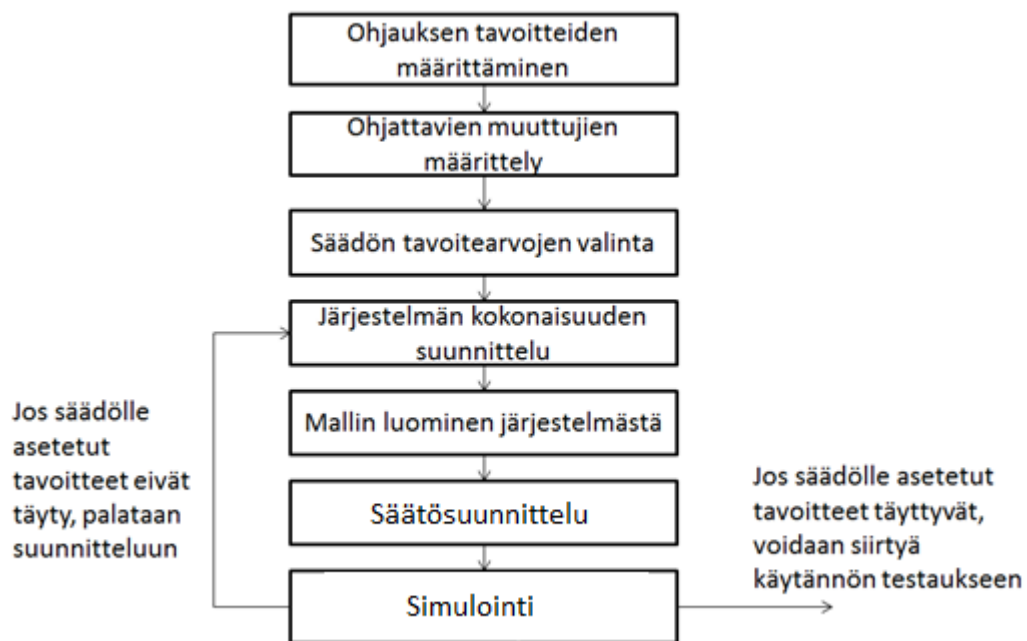


KUVA 4 Analoginen ja digitaalinen servojärjestelmä

2.3 Säätösuunnittelu

Servojärjestelmän säätösuunnittelussa voidaan käyttää kaavion 1 mukaista suunnittelumallia. Ensimmäisenä määritetään mitä varsinaisesti halutaan tehdä, esimerkiksi ajaa

moottoria tietyllä nopeudella. Seuraavaksi määritetään mitä muuttujia ohjauksessa käytetään, esimerkiksi moottorin jännite. Säädön tavoitearvojen valinnassa päätetään missä rajoissa järjestelmä saavuttaa halutun tilan, esimerkiksi kuinka nopeasti moottori asettuu tietylle nopeudelle. Kun tiedetään mitä halutaan, tarvitsee suunnitella järjestelmä kokonaisuus, eli mitä kaikkea järjestelmään kuuluu, esimerkiksi moottori ja mahdolliset kuormat. Järjestelmästä luodaan matemaattinen malli, jonka perusteella pystytään määrittämään ohjaukseen käytettävät funktiot ja eri säätötapojen tarve. Ohjaukseen käytetyn funktion perusteella lasketaan säätöparametreille arvot ja analysoidaan niiden pohjalta systeemin käytöstä simulointien avulla. Jos parametreilla saavutetaan säädölle asetetut tavoitteet, voidaan siirtyä käytännön testaukseen ja säätimen implementointiin, muulloin palataan takaisin suunnitteluun. Säätösuunnittelu on luonteeltaan iteroivaa.



KAAVIO 1 Servojärjestelmän säätösuunnittelumalli (R. Dorf & R. Bishop. 2010.)

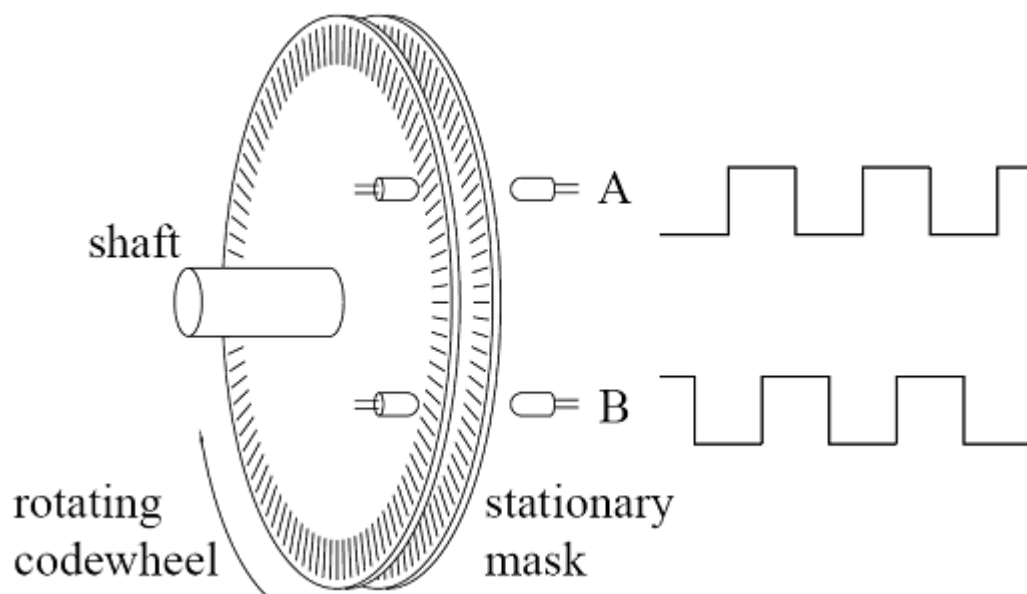
2.4 Sähköservojärjestelmän komponentit

Sähköservojärjestelmä vaatii toimiakseen tarpeelliset komponentit; järjestelmän toimintaa tarkkailevat anturit, sähkömoottorin, sekä ohjausjärjestelmän.

2.4.1 Anturit

Servojärjestelmän mittauslaitteina voi olla esimerkiksi pulssiantureita, potentiometrejä tai takometrejä, sen mukaan mitä halutaan mitata ja kuinka tarkasti. Pulssianturit ovat suosittuja digitaalianpureita kohtuullisen hintansa, tarkkuutensa, monipuolisuutensa ja helpon kytkentänsä vuoksi (Airila. 1999. 4.12). Pulssianturin avulla selviää moottorin pyörimisnopeus, suunta, sekä asema. Takometrillä voidaan mitata vain pyörimisnopeutta, se ei itsessään pysty ilmoittamaan pyörimissuuntaa, jonka vuoksi sen käyttökohteena on yleisesti vain nopeustietoa tarvitsevat järjestelmät.

Kuvassa 5 on esitettyä 2 pulssisen enkooderin toimintaperiaate. Anturit reagoivat 90 asteen vaihesiirrossa toisiinsa nähden. Vaihesiirron suunnasta voidaan päätellä pyörimissuunta. Ohjausjärjestelmä tai sen erillinen laskurikortti laskee pulssien nousevia reunoja ja antaa jatkuvan asema- ja suuntatiedon. (Airila. 1999. 4.12)



KUVA 5. Enkooderin toimintaperiaate (Northwestern mechatronics)

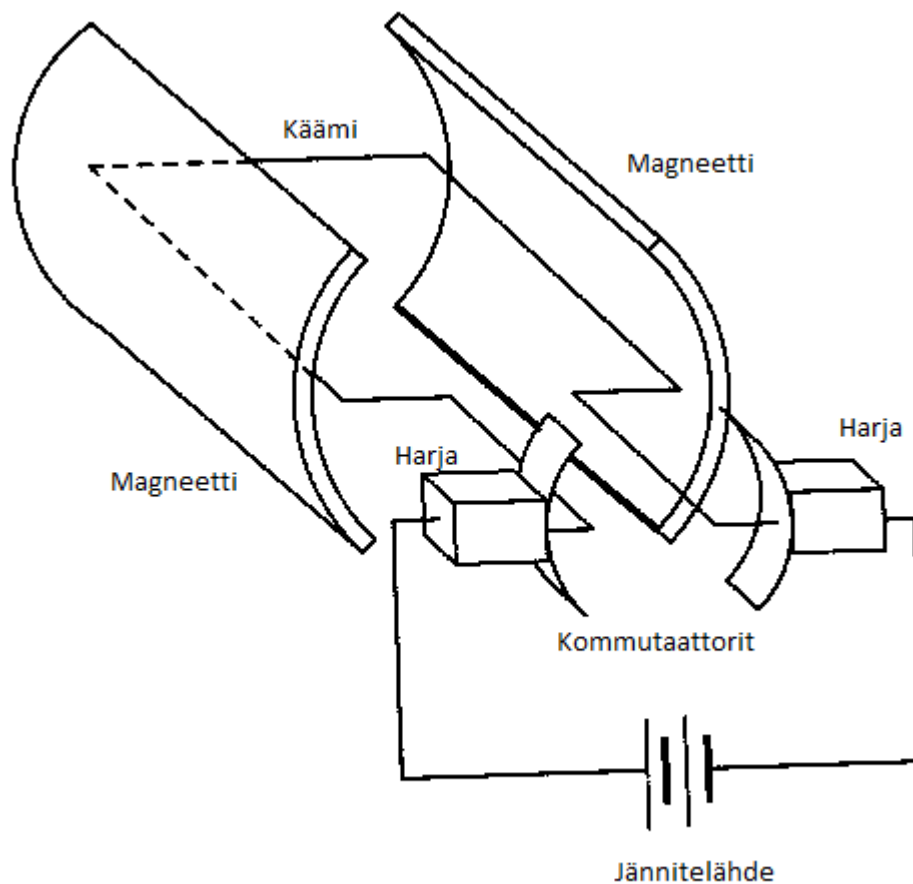
2.4.2 Servomoottori

Servomoottoreina voidaan käyttää sekä tasasähkö, eli DC-, että vaihtosähkö, eli AC-moottoreita.

DC-moottorit ovat toistaiseksi yleisimpiä servomoottoreita hyvien servokäyttöominaisuuksien ansiosta (Airila. 1999, 5.4). Tasavirtamoottoreita käytetään paljon, koska ne ovat halpoja ja niissä on suhteellisesti korkea vääntömomentti. Tasavirtamoottoria voidaan käyttää myös suoraan generaattorina, jolloin sitä pyörittämällä saadaan sähkövirtaa (Inkinen & Tuohi. 1999).

Tasavirtamoottoreissa pyörivässä osassa, eli roottorissa, on käämi, johon johdetaan sähkövirtaa. Kiinteässä osassa, eli staattorissa, on moottorityypistä riippuen joko kestopagneetti tai käämi. Kestomagneettista staattoria käytetään pienissä moottoreissa, isommissa moottoreissa staattorina käytetään käämiä, jolla luodaan sähköä avulla magneettikenttä. Tasavirtamoottoreita voi olla harjattomia tai harjallisia. Harjallisessa moottorissa roottorin käämiin johdetaan sähkövirtaa kommutaattoreihin kytkeytyvien harjojen avulla, harjattomassa hyödynnetään taajusmuuttajaa. Harjaton moottori on huomattavasti huoltovaipaampi kuin harjallinen moottori. (Airila. 1993. 5.5)

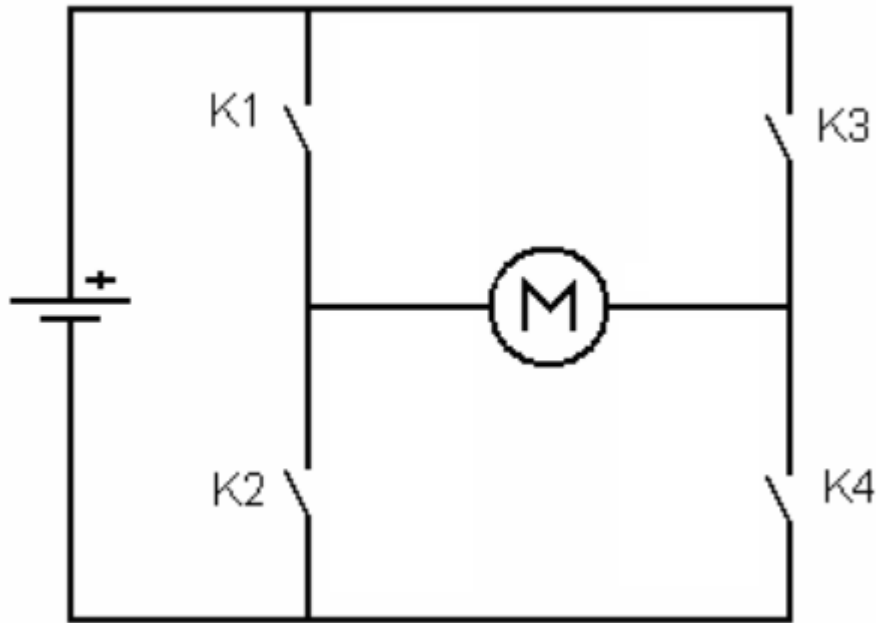
Kuvassa 6 on esitettyä yksinkertaistettu kuva harjallisen tasavirtamoottorin toimintaperiaatteesta. Kestomagneetit muodostavat moottorin staattorin. Roottorin muodostavat käämi ja kommutaattorit. Hiiliharjojen kautta kommutaattoreihin johdettu sähkövirta muodostaa käämiin magneettikentän, joka on erisuunnassa kestopagneettien luomaa kenttää vastaan, jolloin käämi pyrkii pyörittämään roottoria, kunnes magneettikentät ovat samansuuntaiset (Airila. 1993. 5.5). Roottorin pyöriessä, kommutaattorit kytkeytyvät hiiliharjoihin päinvastaisella tavalla, jolloin magneettikentät ovat taas erisuunnassa ja pyörimisliike jatkuu.



KUVA 6 Tasavirtamoottorin yksinkertaistettu toimintaperiaate

Tasavirtamoottorin ominaisuuksia hyödynnetään esimerkiksi askelmoottorissa, jossa käämin kytkentä ei muutu automaattisesti moottorin pyöriessä, käämien kytkentää pitää muuttaa erikseen ohjauspulsseilla, jolloin roottori kääntyy aina tietyn verran. Jos askelmoottoria käytetään oikealla toiminta-alueella, lähetetyistä ohjauspulsseista voi laskea roottorin tarkan aseman ja pyörimisnopeuden ilman asema- tai nopeustakasinkytöntä (Airila. 1999. 5.28). Askelmoottoreiden käytön haittapuolena ovat ylikuormitustilanteet, joissa se ei välttämättä pysty suorittamaan ohjauspulssin mukaisia askeleita.

Tasavirtamoottorin pyörimissuunnan ohjauksessa käytetään yleisesti H-siltakytkentää. H-siltakytkennän havainnekuva on esitettyä kuvassa 7. H-silta hyödyntää neljää kytkintä, jotka ovat yleensä releitä tai transistoreita. Kytkinten eri asentojen vaikutus moottoriin on esitettyä taulukossa 1. Kaikkien kytkinten sulkeminen aiheuttaa oikosulun, joka voi rikkoa moottorin. (Modularcircuits: h-bridges the basics. 2011).



KUVA 7 Tasavirtamoottorin H-siltaohjauksen havainnekuva

TAULUKKO 1 H-kytkennän kytkentävaihtoehdot.

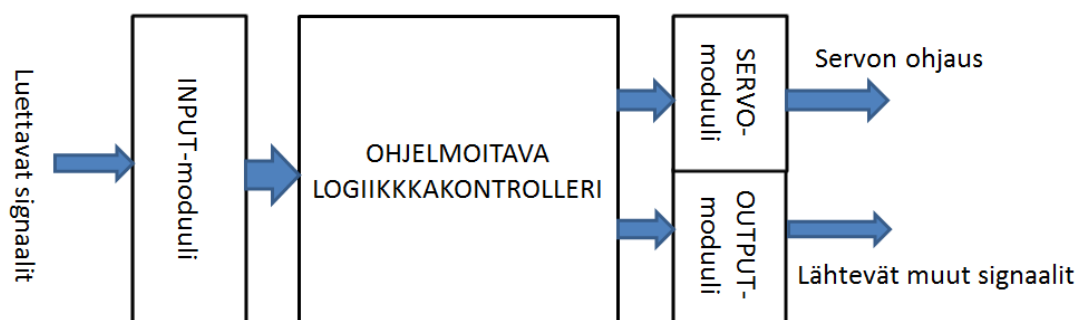
K1	K2	K3	K4	Toiminta
1	1	1	1	Oikosulku
0	1	0	1	Jarru
0	0	0	0	Vapaa
1	0	0	1	Pyörimissuunta 1
0	1	1	0	Pyörimissuunta 2

AC-servomoottorit toimivat nimensä mukaisesta vaihtovirralla. AC-moottoreiden perus toimintaperiaate on DC-moottoreiden kanssa samanlainen, eli roottorin ja staattorin välisen magneettikentän muuttaminen. AC-moottorissa käämeihin syötetään sinimuotoista vaihtojännitettä, jolla moottorin pyörimisnopeutta ja momenttia säädetään. Verrattuna DC-moottoreihin, AC-moottoreiden ohjaukseen tarvitaan enemmän komponentteja, kuten esimerkiksi taajuusmuuttajia. AC-moottorit jaetaan kahteen eri tyyppiin; tahtimoottorit ja epätahtimoottorit. Epätahtiperiaatteella toimivien AC-moottoreiden ominaisuudet ovat epälineaarisia, mikä hankaloittaa moottorin sovittamista takaisinkytkettyihin järjestelmiin (Airila. 1999. 5.27). AC-servomoottoreita käytetään teollisuudessa yleensä servojärjestelmissä, joissa tarvitaan suurta moottoria, muissa tapauksissa DC-servo on käytännöllisempi sen suhteellisesti paremman väännön ja helpomman ohjauksen vuoksi.

2.4.3 Ohjausjärjestelmät

Ohjausjärjestelmät ovat mekatronisten laitteiden aivot. Ohjausjärjestelmä käsittelee antureiden signaalitiedon ja sen perusteella antaa ohjaussignaaleja toimilaitteille. Teollisuudessa ohjausjärjestelminä käytetään ohjelmoitavia logiikkakontrollereita, eli PLC:tä. Ohjelmoitavat logiikat ovat mikroprosessorin sisältäviä tietokoneita, joihin on tarjolla runsaasti eri tarkoituksiin valmistettuja laajennusosia. Ohjelmoitavat logiikat käsittelevät dataa syklisessä ohjelmakierrossa ja niitä voidaan ohjelmoida todella monipuolisesti.

Ohjelmoitavalla logiikalla voidaan hallita esimerkiksi useampaa servotoimilaitetta, antamalla kontrollerin ohjelman perusteella laskettu asetusarvo servojärjestelmälle. Ohjelmoitaviin logiikoihin on saatavilla erillisiä servovahvistimia, joilla servokäyttöjä ohjataan. Kuvassa 8 on esimerkki siitä, miten ohjelmoitavalla logiikkakontrollerilla ja servomoduulilla ohjataan järjestelmän toimintaa. Logiikan tulopuolelle, eli INPUT-moduuliin tulee antureilta signaaleja, logiikka käsittelee signaalit ja muuttaa ohjaussignaaleja tarpeen mukaisesti. Servomoduulille annetaan erillinen asetusarvo ja moduuli itsessään hoitaa servon ohjausta.



KUVA 8 Esimerkki logiikkakontrollerin ja servomoduulin toiminnasta

Muulla kuin teollisuudessa, käytetään yleisesti yksinkertaisia mikrokontrollereita. Mikrokontrollerit ovat yksinkertaisia tietokoneita, jotka on sijoitettu yhdelle ainoalle mikropiirille. Mikrokontrollerit yleisesti sisältävät seuraavia komponentteja:

- mikroprosessori (CPU),

- mikroprosessorin työmuisti, eli RAM,
- mikroprosessorin ohjelmamuisti, eli ROM
- input/output-portteja,
- ajastimia ja laskureita
- keskeytystoimintoja
- analogiamuuntimia
- sarjaliitäntä
- oskillaattoriipiirejä. (CircuitsToday: Basics of Microcontrollers 2011.)

Mikrokontrollereilla voidaan toteuttaa monenlaisia sovellutuksia, ohjata esimerkiksi erilaisia robotteja tai kodinkoneita. Mikrokontrolleiden ohjelmointiin käytetään yleisesti C ja C++ ohjelmointikieliin perustuvia ympäristöjä.

Mikrokontrollereita on laaja valikoima ja niiden etuna on pieni koko ja edullisuus. Mikrokontrollerit on tarkoitettu toimimaan lähinnä ohjaussignaalien tuottamiseen, eli niillä ohjattavat laitteet tarvitsevat lähes aina erillisen virtalähteen, koska mikrokontrollereiden lähtösignaalien virta ei ole kovin suuri. Myös tulosignaalit pitää suunnitella niin että niiden virta pysyy pienenä ja jännite tarpeeksi alhaisena.

Nopeasti suosiota kasvattava mikrokontrolleri on Arduino. Arduino on avoimeen lähdekoodiin perustuva mikrokontrolleri, josta on erilaisia versioita tarpeen mukaan. Arduinon yleistymisen syinä ovat:

- halpa hinta,
- sen kehitysympäristö ei ole käyttöjärjestelmärajotteinen,
- kehitysympäristö on helppokäyttöinen, mutta kuitenkin monipuolinen,
- avoin lähdekoodi, jonka myötä ohjelmoinnissa voi hyödyttää esimerkiksi C++ kirjastoja. (Arduino: Introduction. 2017).

Arduinon ohjelmointiin on oma ohjelmointialusta Arduino IDE. Arduinon ohjelmointi perustuu hyvin vahvasti C ja C++ ohjelmointikieliin, joten sen käyttäminen on hyvin helppoa.

3 LAITTEEN SUUNNITTELU

3.1 Laitteen vaatimukset

Laitteen suunnittelun lähtökohdiksi asetettiin muutama vaatimus joiden pitää täyttyä. Vaatimuksia olivat

- ohjauksen tulee olla mahdollista Matlabin + Simulinkin ja Arduinon kautta
- moottorin avulla tulee pystyä liikuttamaan 0-2 kg hyötymassaa lineaarisesti vaaka- sekä pystysuunnassa,
- laitteen pitää olla riittävän pieni käytön ja säilytyksen helpottamiseksi,
- moottorin liikettä pitää pystyä vastustamaan on/off tyylisellä vastamomentilla, häiriövasteen tutkimiseksi,
- mahdollisimman paljon laitteen rungon osista tulee suunnitella laserleikkaamalla valmistettaviksi,
- pyrkimys modulaarisuuteen,
- liikealue n. 200 mm-300 mm,
- n . 40 mm/s liikenopeus.

3.2 Komponenttien valinta

Suunnittelun kannalta tärkeintä oli valita millä tavalla lineaariliike toteutetaan. Vaihtoehtoina olleista hihnavetoisesta kelkasta ja liikeruuvista selvästi halvempi vaihtoehto on liikeruuvi, joten se otettiin liikkeen toteutustavaksi. Liikeruuvilla toteutetun lineaariliikkeen nopeuteen vaikuttaa ruuvin nousu, eli kuinka paljon kierre nousee kierrosta kohden, sekä sen pyörimisnopeus.

Komponenttien valintaan vaikutti laitteelle asetettu budjetti sekä osien saatavuus ja toimitusajat, jolloin valikoima pieneni jonkun verran. Suunnittelun tuloksena syntynyt komponenttilista on esitettyä kokonaisuudessaan taulukossa 2.

Taulukko 2. Komponenttilista

Komponentti	Tarkennus	Määrä
12V DC-moottori	GM37-520-10000-30+ZJ	1
Kontrolleri	L298N Dual H-bridge	1
Liikeruuvi	8mm halkaisija, 8mm nousu	1
Liikeruvin mutteri	8mm reiällä	1
Johdetanko	350mm pitkä	2
Johdepukki	8mm reiällä	4
Johdelaakeri	8mm reiällä	2
Spiraalikytin	6mm-8mm	1
Spiraalikytin	5mm-8mm	1
Laakeripukki	8mm reiällä	2
Jarrumoottori	Wantai 42BYGHW208	1

3.2.1 Moottori ja liikeruuvi

Moottorin valintaan vaikutti tarvittava vääntömomentti ja kierrosnopeus. Koska lähtövaatimuksissa ei määritetty kuinka paljon massaa laitteen pitää pystyä liikuttamaan, käytetään laskuissa oletuksena 0-2 kilogrammaa. Liikeruuviksi valitaan 8 mm halkaisijaltaan oleva trapetsikierreruuvi, jonka nousu on 8 mm/kierros.

Valitun moottorin datalehti on esitetty liitteessä 1.

Laskuilla varmistetaan, että moottori täyttää vaatimukset. Muutetaan moottorin ilmoitettu pyörimisnopeus RPM muodosta muotoon 1/s.

$$n = \frac{N}{60s} = \frac{320}{60s} \approx 5,333/s$$

Lasketaan lineaariliikkeen nopeus v liikeruuvien nousun (x) ja moottorin kierrosnopeuden (n) avulla.

$$v = n \times x = \frac{5,333}{s} \times 8mm \approx 42,7 \frac{mm}{s}$$

42,7 mm/s, eli 4,27 senttimetriä sekunnissa on riittävän ripeä lineaariliikkeen nopeus. Lasketaan oletetun maksimimassan, 2 kilogramman, aiheuttama voima

$$F = m \times g = 2 \text{ kg} \times 9,81 \frac{\text{m}}{\text{s}^2} \approx 19,6 \text{ N}$$

Lasketaan massan nostamiseksi liikeruvilla tarvittava vääntömomentti (T_{nosto}), liikeruuvien hyötysuhteeksi (μ) valitaan yleisesti käytetty 0,2. Liikeruuvien halkaisija (d) on 8 mm.

$$T_{\text{nosto}} := \frac{F \cdot d}{2} \left(\frac{x + \pi \mu \cdot d}{\pi \cdot d - \mu \cdot x} \right)$$

$$T_{\text{nosto}} := \frac{19,6 \text{ N} \cdot 8 \text{ mm}}{2} \left(\frac{8 \text{ mm} + \pi 0,2 \cdot 8 \text{ mm}}{\pi \cdot 8 \text{ mm} - 0,2 \cdot 8 \text{ mm}} \right) = 0,043 \text{ N} \cdot \text{m}$$

Liitteessä 1 olevassa moottorin datalehdessä ilmoitetaan moottorille muutama eri vääntömomentti eri kierrosnopeuksilla. Laskettu nostoon tarvittava vääntömomentti 0,043 Nm on selvästi alempi kuin moottorille ilmoitetut vääntömomentit, joka varmistaa, että valittu moottori jaksaa liikuttaa tarpeeksi massaa. Laskuissa ei oteta huomioon kitkavoimia, koska ne oletetaan niin pieniksi, ettei niillä ole vaikutusta suunnittelun.

Moottorin suurin käyttöjännite on 12 voltia. Valitussa moottorissa tulee myös mukana kiinnitysrauta. Moottorissa on myös integroituna enkooderi, jonka avulla saadaan pulssitietona tietää moottorin pyörimien kierrosten määrä.

Liikeruuvien pituudeksi valitaan 300 mm, jolloin laakeroinnin ja kytkimien lisäämisen jälkeen liikealueeksi jää 250 mm, joka sopii hyvin suunnittelulle asetettuihin vaatimuksiin.

3.2.2 Vastamomentti

Vastamomentin luomiseksi hyödynnetään askelmoottoria, jonka käämejä oikosulkemalla saadaan luotua erilaisia pitomomentteja. Ilmiö perustuu siihen, että moottorin roottorin pyöriessä, muodostuu sähkövirtaa käämeihin, jotka yhteen kytkettyinä muodostavan sellaisen magneettikentän joka pyrkii vastustamaan staattorin liikettä. Käytettävä askelmoottori on nelijohtiminen ja sillä saadaan luotua kaksi eri momenttia. Heikompi momentti saadaan luotua yhdistämällä askelmoottorin neljästä johtimesta kolme samaan piiriin, voimakkaampi momentti yhdistämällä kaikki johtimet samaan piiriin. Askelmoottorille ei tuoda ulkoista energiaa, vaan momentti perustuu käämeihin luotuihin oikosulkuihin. Askelmoottorin tarkat tiedot selviävät liitteessä 2 olevasta datalehdessä.

Vastamomenttia tulee voida kontrolloida mikrokontrollerilla tarkkoina ajanhetkinä, jonka vuoksi johtimia yhdistetään kahdella, mikrokontrollerin avulla ohjatulla releellä.

Jatkossa on mahdollista vaihtaa askelmoottorin tilalle esimerkiksi tavallinen DC-moottori, jos tarpeet vastamomentin laadulle tai ohjaukselle muuttuvat.

3.2.3 Johteet, laakerit ja kytkimet

Liikeruuvien lisäksi tarvitaan johteet, joita pitkin liikutettava kelkka pääsee liikkumaan lineaarisesti hyvin pienellä kitkalla. Hinnan kannalta edullisin ratkaisu on johdetangot, jotka eroavat kalliimmista johteista siten, että niissä laakeri pääsee pyörimään vapaasti tangon ympäri. Johdetangon halkaisijaksi valitaan sama halkaisija kuin liikeruuvilla, eli 8 mm. Laitteeseen tulee kaksi johdetankoa, yksi liikeruuvien molemmille puolille. Laitteeseen toimisi myös pelkästään yhdellä johdetangolla, mutta kahdella saadaan varmistettua laitteen vakaus kun yritetään liikuttaa suurempia massoja.

Johdetankoon tulee johdelaakerit, joihin liikutettava kelkka kiinnitetään. Liikeruuvien molempiin päihin laitetaan 8 mm laakeripukit, jolloin ruuvi pysyy vakaana ja pyörimistä haittaava kitka hyvin pienenä. Tanko kiinnitetään laitteen runkoon johdepukeilla. Kuvasssa 9 on esitettyä 8 mm johdetangot, johdelaakerit sekä johdepukit.



KUVA 9 Johdetangot, johdelaakerit sekä johdepukit

Moottori ja liikeruuvi yhdistetään 6 mm/8 mm spiraalikytkimellä ja vastamomenttimoottori ja liikeruuvi 5 mm/8 mm spiraalikytkimellä. Spiraalikytkin suojelee moottoria ja muita komponentteja joustamalla ja liian suuressa kuormituksessa hajoamalla ennen kalimpien osien rikkoontumista. Spiraalikytkimen havainnekuva on esitettynä kuvassa 10.



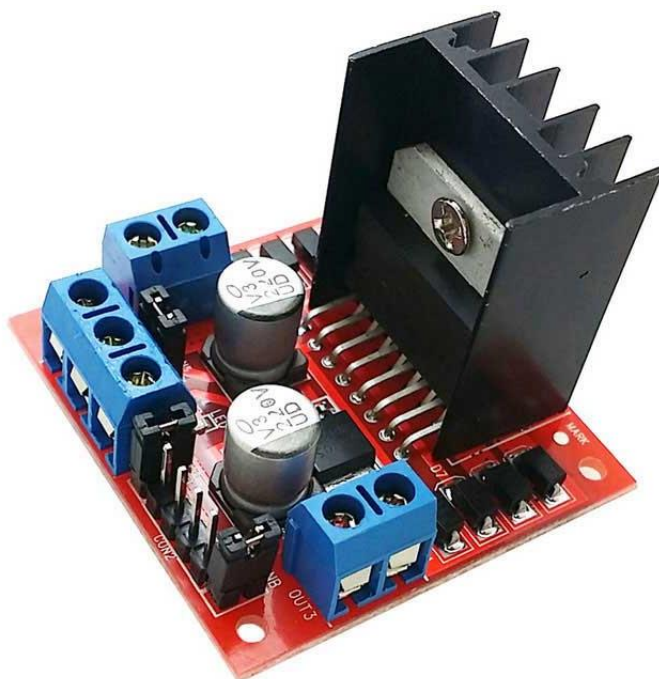
KUVA 10 Spiraalikytkin

Kelkan ajo rajoja vasten estetään rajakytkimien avulla. Kahdella päädyissä olevilla rajakytkimillä katkaistaan suoraan moottorin virta, yhtä käytetään signaalina kotiasemaan ajossa.

3.2.4 Moottorin kontrolleri

Moottorin ajamiseksi hyödynnetään H-silta kytkentää. H-siltaa hyödyntäviä valmiita kontrollerimoduuleja on saatavilla runsaasti, yleisin moduuli 12v voltin käytöissä on L298N-moduuli. H-silta kytkennässä voidaan helposti ja erittäin nopeasti muuttaa moottorin pyörimissuuntaa. Moottorin nopeutta voidaan säätää mikrokontrollerin pulssinleveysmodulaation, eli PWM:n avulla.

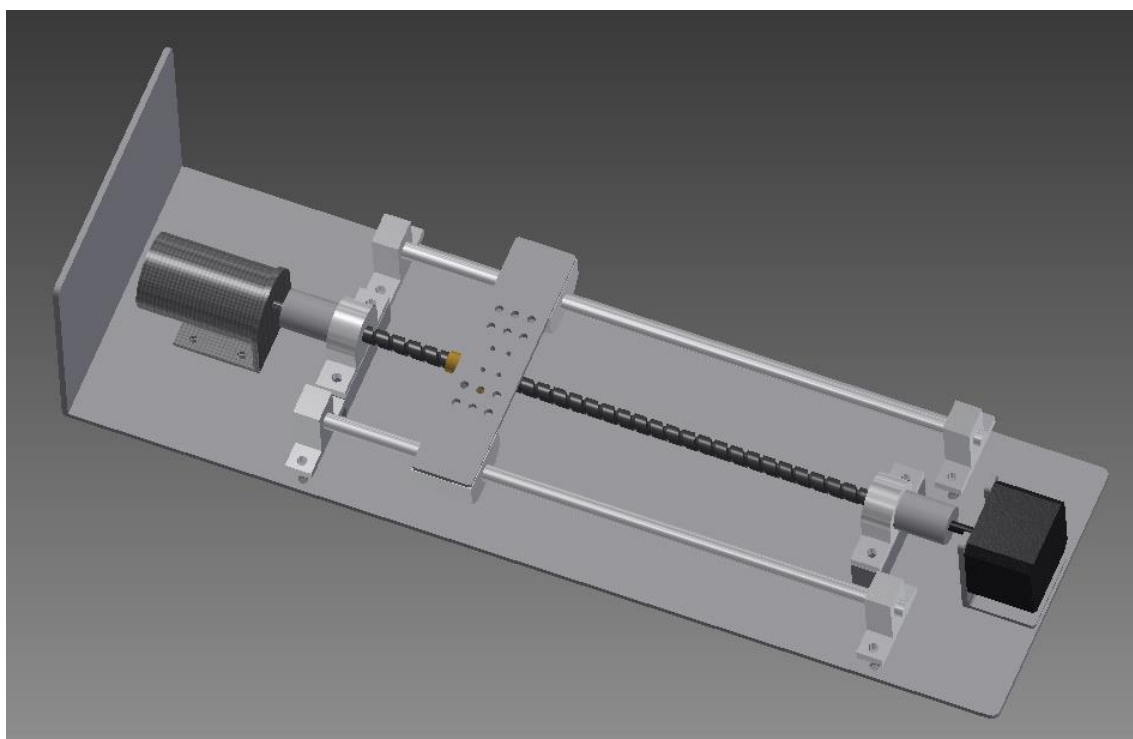
Valitun moottorin virta tehokkaimmalla ajolla on n. 1,2 ampeeria, L298N-moduulin suurin sallittu jatkuva virta on 2 ampeeria, eli L298N-moduuli on sopiva ja edullinen valinta moottorin ohjaamiseen. Moduuli on esitettyinä kuvassa 11. Valitulla kontrollerilla on mahdollista ohjata samaan aikaan kahta DC-moottoria tai yhtä askelmoottoria. Laitteeseen tulee vain yksi DC-moottori, mutta kahden moottorin ajon mahdollistava kontrolleri on hinnaltaan samanlainen kuin pelkän yhden moottorin kontrolleri, sekä potentiaali lisätä toinen moottori lisää laitteen muokattavuutta ja modulaarisuuta.



KUVA 11 H-silta kontrolleri

3.3 Mekaniikkasuunnittelu

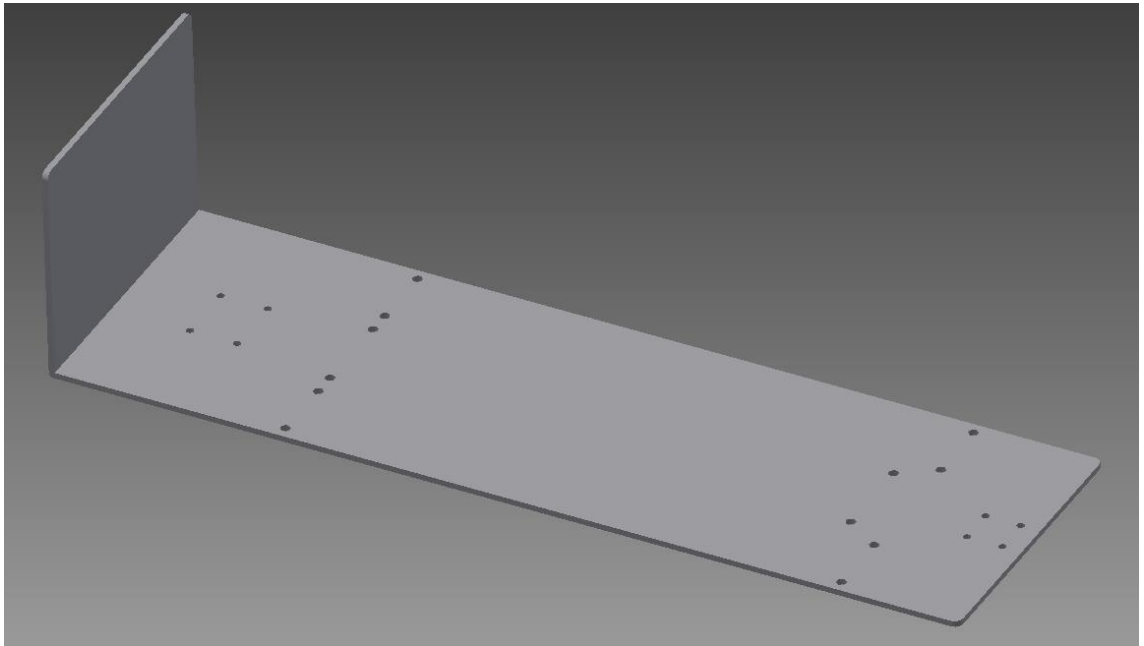
Mekaniikkasuunnitteluun vaikutti valittujen komponenttien koko ja kiinnitystavat. Lähtökohtana oli, että runko sekä muut komponenttien kiinnityspalat valmistetaan laserleikkaamalla levystä. 3D-mallit on piirretty Autodesk Inventor Professional 15 ohjelmalla. 3D-mallinnettu kuva laitteen kokoonpanosta on esitetty kuvassa 12. Kokoonpanokuva erikseen merkattuine osineen on liitteessä 3.



KUVA 12. 3D-malli laitteen kokoonpanosta

3.3.1 Runko

Laitteen vaatimuksissa oli tarve liikuttaa massaa tarvittaessa sekä vaaka- että pystysuunnassa, jonka vuoksi suunnittelussa päädyttiin runkoon, jonka myötä koko laitteen voi asettaa vaaka- tai pystyasentoon tarpeen mukaan. Laserleikattavaan runkolevyyn on piirretty kiinnitysreiät komponenteille. Levy kantataan 90° asteen kulmaan. Runkolevyn 3D malli on esitetty kuvassa 13. Mitoitettu 2D-malli on liitteessä 4.



KUVA 13. Rungon 3D-malli

3.3.2 Jarruna toimivan askelmoottorin teline

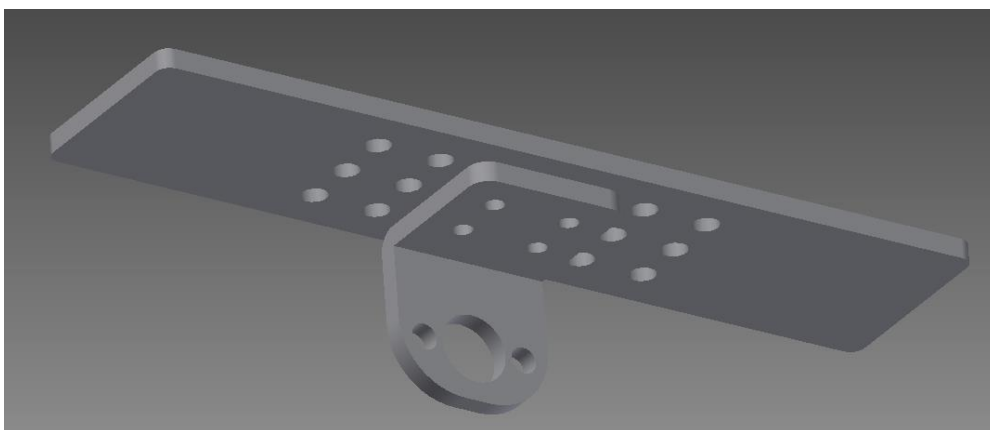
Vastamomenttina toimiva askelmoottori kiinnitetään runkoon erillisellä telineellä, teline on suunniteltu laserleikattavaksi ja kantattavaksi 90° asteen kulmaan. Telineessä olevan askelmoottorin korkeutta pystyy säätämään. Telineen 3D malli on esitetty kuvassa 14. Mitoitettu 2D-malli on liitteessä 5.



Kuva 14. Askelmoottoritelineen 3D-malli

3.3.3 Kelkka

Kelkka yhdistää trapetsikierremutterin sekä johdelaakerit yhdessä liikkuvaksi kokonaisuudeksi. Kelkka koostuu kahdesta osasta, mutterin kiinnitysosasta ja päälliosasta, jossa on reikiä, joiden avulla voi kiinnittää kelkkaan tarvittavan kuorman. Molemmat osat valmistetaan laserleikkaamalla, kiinnitysosa kantataan 90° asteen kulmaan. Kelkan 3D-malli on esitettynä kuvassa 15. Mitoitettut 2D-mallit ovat liitteissä 6 ja 7. Suunnitteluvaiheessa ei ollut saatavilla johdelaakereiden kiinnitysreikien mitoituksia, joten kelkkaan ei ole piirretty kaikkia kiinnitysreikiä.



Kuva 15. Kelkan 3D-malli

3.3.4 Korokepalat

Liikervuvin korkeus runkoon nähden määrittyy moottorin kiinnitysrangan ja moottorin akselin myötä, minkä takia muita komponentteja pitää saada nostettua tiettyyn korkeuteen

Jarruna käytettävän askelmoottorin telineeseen on tehty kiinnityspultteja varten pitkät urat, joka antaa vapauden määrittää askelmoottorin akselin korkeus samalle tasalle kuin ajomoottorin akselin korkeus.

Liikervuvin laakeripukkien ja johdepukkien alle tarvitaan juuri oikean korkuiset korokepalat.

3.4 Sähkösuunnittelu

Laitteen sähkökuvat on piirretty CADS ELECTRIC 17 ohjelmalla. Käytettävät sähkökomponentit ovat esitettynä taulukossa 3. Sähkökaavio kokonaisuudessaan on liitteessä 8.

TAULUKKO 3. Sähkökomponentit

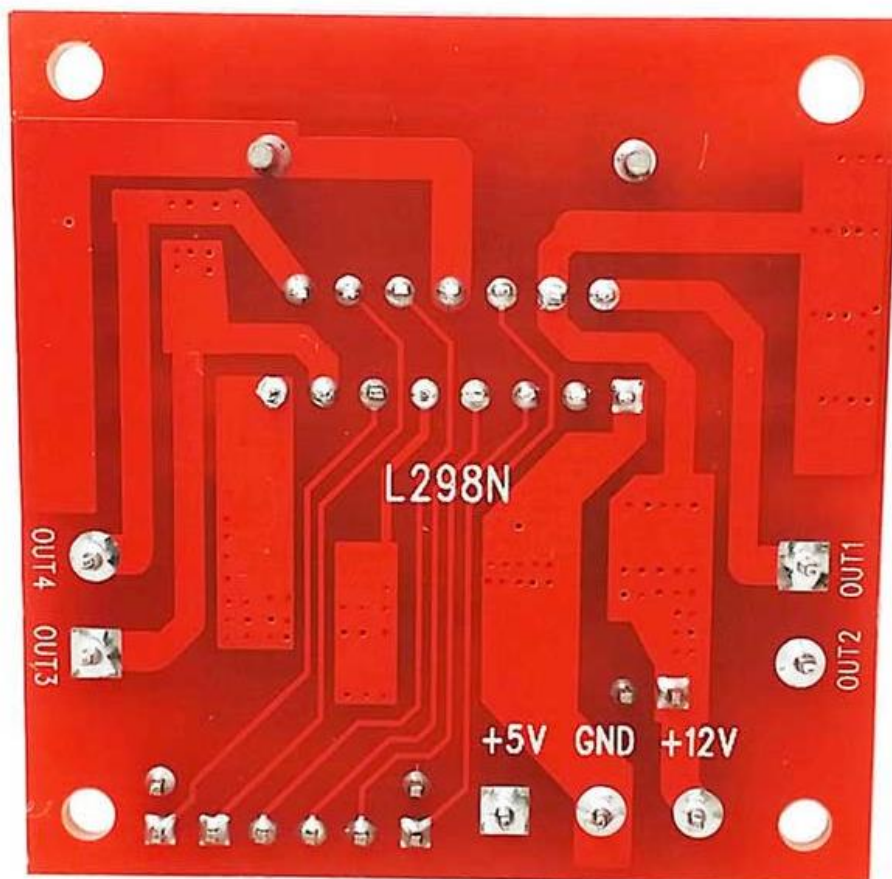
Komponentti	Lisätietoja	Määrä
Rele	5V, NO	2
Vastus	8,2 kΩ	2
Sulake	2 A	1
Rajakytkin	NC	3

3.4.1 L298N-moduuli ja moottori

Moottorin kontrollerissa on 13 pinniä:

- 12 voltin tulojännite
- 5 voltin referenssijännite
- GND, eli maa
- 4 pinniä H-sillan ohjaukseen
- 2 enable-pinniä, ENABLE A ja ENABLE B, joiden avulla voidaan säätää kontrollerin lähtöjänniteitä
- 4 OUT-pinniä, joilla voidaan ohjata kahta moottoria, tai yhtä askelmoottoria.

Kontrollerin takaosa, josta sen pinnit näkee, on esitetty kuvassa 16. Moottori kytketään pinneihin OUT 1 ja OUT 2, Moottorin johtojen kytkennässä ei varsinaisesti ole väliä miten päin johtimet kytketään, koska moottorin pyörimissuunta kuitenkin säädetään kontrollerin H-sillan avulla.



KUVA 16. L298N H-silta

3.4.2 Arduino

Laitteen vaatimuksissa oli, että sitä tulee pystyä ohjaamaan Matlabin ja Simulinkin kautta. Matlabin saa kirjaston, jonka avulla sen saa yhdistettyä Arduino mikrokontrolleriin, joten Arduinon valinta ohjauslaitteeksi oli järkevä.

Laitteeseen ei kiinteästi integroida Arduino-mikrokontrolleria, vaan se kiinnitetään aina erikseen kun laitetta aletaan käyttämään. Arduino kytketään opetuslaitetta käytettäessä myös aina tietokoneeseen USB-yhteydellä, tämän vuoksi Arduinon 5 voltin jännitelähteenä käytetään USB:n kautta tulevaa jännitettä. Laitetta voi käyttää myös muilla mikrokontrollereilla kuin Arduinolla, mutta niitä käytettäessä tulee varmistaa mikrokontrollerin tulojen ja lähtöjen tarkat virta- ja jänniterajat ja tarpeet.

Arduinosta käytetään neljää digitaalilähtöä, kolmea digitaalituloa, sekä yhtä PWM-lähtöä, eli pulssinleveysmodulaatiota.

Digitaalilähtöjä ovat:

- kontrollerin H-siltaa ohjaavat IN1 ja IN2
- releiden K1 ja K2 ohjaus.

Digitaalituloja ovat:

- moottorin enkooderin kaksi pulssituloa
- rajakytkin 3:n signaali

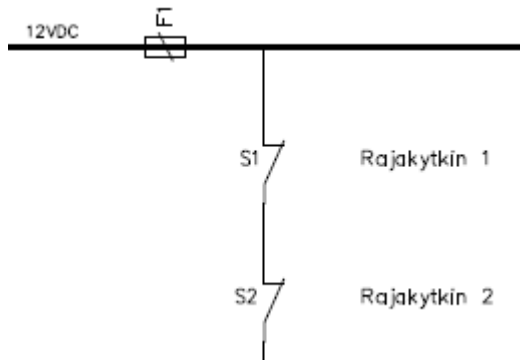
Arduinon digitaalituloja käytettäessä tulee ottaa huomioon huojuntajännite, joka voi aiheuttaa virheellisiä signaaleja. Huojuntajännitteestä päästään eroon käyttämällä pinniin kytkettyä alavetovastusta, joka pitää pinnin jännitteen 0 voltissa, kunnes pinnille annetaan oikea signaali. Alavetovastuksena käytetään 8,2 k Ω vastusta.

PWM-lähtö kytketään L298N-moduulin ENABLE A pinniin. PWM-lähdöllä saadaan syötettyä moduuliin 0 – 5 Voltin jännite, jonka moduuli muuttaa 5 – 12 Voltin jännitteeksi.

3.4.3 Rajakytkimet ja sulake

Moottorin ja kontrollerin suojaamiseksi käytetään 2 ampeerin sulaketta, sulake kytketään 12 voltin jännitelähteen ja kontrollerin väliin.

Laitteen molemmissa päissä on rajakytkin, joilla estetään laitteen ajo rajojen yli ja näin laitteen rikkoutuminen. Molemman pään rajakytkin kytketään NC-muodossa sarjaan 12 voltin jännitelähteen ja moottorin kontrollerin väliin, jolloin rajalle ajo katkaisee virtapiirin ja estää moottorin liikkeen jatkamisen. Rajakytkimien ja sulakkeen kytkentä on esitettyä kuvassa 17.

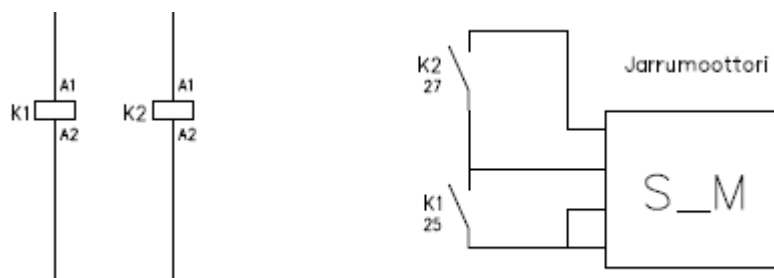


KUVA 17. Sulakkeen ja rajakytkimien kytkentä

Yhdellä rajakytkimellä toteutetaan kotiasemaan ajon mahdollistavan paikkasignaalin tulo Arduinolle. Rajakytkin kytketään NC-muodossa ja se sen kautta ohjataan 5 voltin jännitettä.

3.4.4 Jarrumoottorin kytkentä

Jarruna käytettävän askelmoottorin ohjaus toteutetaan rele-kytkennöillä. Käytetään kahta 5 voltilla toimivaa mikrokontrollerilla ohjattavaa relettä, joiden kärjet kytketään NO-muodossa. Askelmoottorin johtimet 1 ja 2 kytketään suoraan yhteen, yhdistetyt johtimet kytketään kiinni releen K1 kärkeen 1. Releen K1 kärkeen 2 kytketään askelmoottorin johdin 3, johdin kytketään myös kiinni releen K2 kärkeen 1. Releen K2 kärkeen 2 kytketään askelmoottorin johdin 4. Kytkentä on esitettyä kuvassa 18.



KUVA 18. Askelmoottorin kytkentä

4 TESTIOHJELMA

Laitteen toiminnan testaukseen käytetään yksikertaista ohjelmaa, jonka avulla kelkkaa ajetaan tiettyyn paikkaan. Testiohjelma kommentoituna on esitettyä kokonaisuudessaan liitteessä 9. Ohjelma on kirjoitettu Arduino IDE ohjelmaeditorilla. Käytettävien tulojen ja lähtöjen, eli pinnien, määrittelyssä on käytetty Arduino UNO mikrokontrollerin portti-configuraatiota, mikä pitää muistaa jos ohjelmaa ajetaan jollakin muulla mikrokontrollerilla, koska eri tyyppin mikrokontrollereissa on yleensä erikseen määritetyt pinnit keskeytyksille ja pulssinleveysmodulaatiolle.

4.1 Kotiasemaan ajo

Ensimmäisenä ohjelmakierrossa aletaan suorittaa aliohjelmaa *kotiajo*, jolla kelkka ajetaan sen kotiasemaan, eli paikkaan joka on määritetty rajakytkimellä. Aliohjelmassa kelkkaa ajetaan lineaarisuunnassa kotiasemaan päin niin kauan, että rajakytkin vaikuttaa, Kun ohjelma saa signaalin rajakytkimen vaikuttumisesta, poistutaan *kotiajo* aliohjelmasta eikä siihen enää palata ilman että mikrokontrollerin ohjelmasuoritus resetoidaan. Aliohjelmassa hyödynnetään while-silmukkaa, jonka sisäistä ohjelmakiertoa toistetaan, kunnes silmukan toistoehto muutetaan epätodeksi. Ennen aliohjelmasta poistumista muutetaan kelkan paikkatietoa tallentavan muuttujan *paikka* arvo nolaksi, koska kotiasemaan ajossa enkooderin pulssit muuttavat sen arvoa. Kotiasemaan ajo suoritetaan vain kerran, tämä varmistetaan pääohjelmassa olevalla muuttujaan liittyvällä ehdolla, muuttujan arvo muutetaan *kotiasema* aliohjelmassa niin että ehto ei enää täyty.

4.2 Enkooderin lukeminen

Paikan määrittelyyn käytetään moottorin enkooderin pulssitietoja. Enkooderista saadaan ulos kaksi pulssia, joten niistä selviää myös moottorin pyörimissuunta. Moottorin datalehdessä selviää, että enkooderin resoluutio on 330 pulssia jokaista vaihteiston jälkeisen akselin kierrosta kohden.

Arduinon mikroprosessorin ohjelmakierto ei ole riittävän nopea enkooderin pulssien tulkintaan normaalissa ohjelmakierrossa, joten käytetään *attachinterrupt* toimintoa, joka hyödyntää mikrokontrollerin keskeytystoimintoa, jonka avulla mikrokontrolleri reagoi jokaiseen erikseen määritellyn digitaalitulon signaalimuutokseen. Pulssisignaalin muutos siirtää ohjelmakierron suoraan erikseen määritettyyn aliohjelmaan *enkooderi*, jossa muutetaan paikkatiedolle varatun globaalin muuttujan *paikka* arvoa joko suuremmaksi tai pienemmäksi, sen mukaan missä tilassa enkooderin signaalit ovat toisen niistä muuttuessa.

Keskeytystoiminnolla tarkkaillaan vain enkooderin toisen pulssin tilanmuutoksia, koska tällä tavalla pystytään määrittämään pyörimissuunta. Kun tarkkailtava pulssisignaali muuttu, tarkistetaan missä tilassa toinen pulssisignaali on ja sen avulla päätellään mihin suuntaan moottori pyörii. Kun tarkkailtava pulssisignaali muuttuu HIGH tilaan ja toinen pulssisignaali on LOW, tiedetään että moottori pyörii myötäpäivään, jos toinen pulssisignaali on HIGH, tiedetään että moottori pyörii vastapäivään. Kun tarkkailtava pulssisignaali muuttuu LOW tilaan ja toinen pulssisignaali on LOW, tiedetään että moottori pyörii vastapäivään, jos toinen pulssisignaali on HIGH, tiedetään että moottori pyörii myötäpäivään.

4.3 Moottorin ajo

Moottorin ajamiseksi käytetään kahta eri aliohjelmaa, *ajo_eteenpain* ja *ajo_taksepain*. Aliohjelmissa määritetään H-sillan kytkennät kahdella lähdöllä niin, että moottoria ajettaessa toinen lähdöistä on HIGH ja toinen LOW, vaihtamalla tiloja päinvastoin moottorin suunta vaihtuu. Nopeuden säätämiseksi käytetään mikrokontrollerin pulssinleveysmodulaatiota, jolla voidaan syöttää moottorin kontrollerille 0-5 voltin jännitettä 8 bitin resoluutiolla, eli arvoilla 0-255. Nopeussäädön arvo viedään aliohjelmaan pääohjelmasta muuttujana *nopeus*. Testiohjelmassa käytetään arvoa 150, mutta nopeussäädön arvo voidaan tarvittaessa muuttaa, kun lähestytään tavoiteltua paikkaa. Moottorin ajon pysäyttämiseksi käytetään aliohjelmaa *moottorin_vapautus*, jolla kytketään H-sillan kytkennät pois päältä.

Moottorin suunta ja nopeus pääohjelmakierrossa vertaamalla paikkatiedon arvoa tavoitteeksi asetettuun *tavoitepaikka* arvoon. Ohjelmaan määritetään myös *toleranssi* muuttujalla kuinka tarkasti haluttuun paikkaan halutaan ajaa, eli kuinka paljon *paikka* ja *tavoitepaikka* voivat erota toisistaan, ilman että moottoria ajetaan. Mikäli *paikka* on pienempi kuin *tavoitepaikka* otettuna huomioon *toleranssi*, ajetaan moottoria eteenpäin aliohjelmalla *ajo_eteenpain*, mikäli *paikka* on suurempi kuin *tavoitepaikka* otettuna huomioon *toleranssi*, ajetaan moottoria taaksepäin aliohjelmalla *ajo_taa**ks**epain*. Jos *paikka* ja *tavoitepaikka* ovat *toleranssi* huomioon ottaen tarpeeksi lähellä, vapautetaan moottori *moottorin_vapautus* aliohjelmalla.

5 YHTEENVETO JA POHDINTA

Työssä saatiin suunniteltua kokonaisuudessaan laite, joka vastaa alussa asetettuja tavoitteita. Työtä hankaloitti selvästi nopea aikataulu, jonka vuoksi kaikki suunnitellut ratkaisut eivät välttämättä osoittautu parhaimmaksi valinnaksi laitteen rakentamisen yhteydessä.

Komponenttivalinnoissa piti tehdä kompromisseja komponenttien saatavuuden ja hinnan vuoksi, komponentit myös vaikuttivat hyvin merkittäväällä tavalla muihin suunnittelukoh-teisiin. Valitut komponentit kuitenkin täyttävät suunnittelun aluksi määritellyt vaatimuk-set laitteen toiminnalle, mutta vasta varsinaisessa testauksessa selviää, kuinka tarkasti esi-merkiksi moottorin tiedot pitivät paikkansa.

Tavoite liikuttaa 2 kilogramman hyötymassaa tietyllä nopeudella asetti haastetta löytää sopiva moottori, jonka hinta ei olisi liian korkea. Valitsemalla lineaariliikkeen toteutus-muodoksi liikeruuvi, päästiin laskemaan liikutettavan hyötymassan ja liikeruuvin nousun avulla moottorilta vaadittava vääntömomentti sekä kierrosnopeus.

Laitteen ohjaus onnistuu tarvittaessa pelkän Arduino-mikrokontrollerin kautta, tai Matla-bin ja Simulinkin kanssa kommunikoivan Arduinon avulla. Häiriövasteen tutkimiseksi moottorille valittiin vastamomentin tuottamiseen askelmoottori, jonka avulla saadaan luotua kaksi erisuuruista vastamomenttia. Askelmoottorin käyttö vastamomentin tuotta-miseen vaikuttaa järkevältä ratkaisulta, mutta suunnitelman modulaarisuuden vuoksi as-kelmoottorin tilalle voidaan vaihtaa tarvittaessa myös muunlainen ratkaisu.

Pyrkimys suunnitella mahdollisimman moni laitteen rungon osista valmistettavaksi laser-leikkaamalla, onnistui hyvin, kuudesta erilaisesta osasta neljä pystytään valmistamaan la-serleikkaamalla. Suunnittelemalla laite käytettäväksi kahdessa eri asennossa, saavutettiin tavoite liikuttaa massaa sekä pysty- että vaakasuunnassa. Laitteen kokokin pysyi tavoit-teen mukaisesti suhteellisen pienenä.

Laitteen avulla voidaan servotekniikan opetuksessa jatkossa havainnollistaa erilaisten säätötapojen vaikutusta DC-sähköservojärjestelmän käyttäytymiseen sekä tutkia miten si-mulointi ja käytännön järjestelmä eroavat toisistaan säädön viimeistelyssä.

LÄHTEET

Airila, M. 1993. Mekatroniikka. Helsinki: Hakapaino Oy.

Arduino. 2017. Arduino homepage. Luettu 20.4.2017 <http://arduino.cc/en/Reference/HomePage>

Circuitstoday. 2011. Basics of Microcontrollers. Luettu 28.4.2017. <http://www.circuitstoday.com/basics-of-microcontrollers>

Fonselius, J., Rinkinen, J. & Vilenius, M. 1998. Servotekniikka. Helsinki: Oy Edita Ab.

Dorf, R & Bishop, R. 2010. Modern Control Systems.

Inkinen, P., Tuohi, J. 1999. Momentti 1, insinöörifyysiikka. Keuruu: Otava.

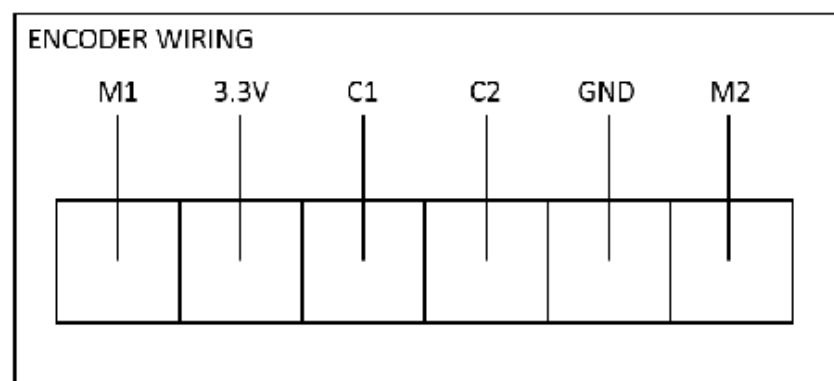
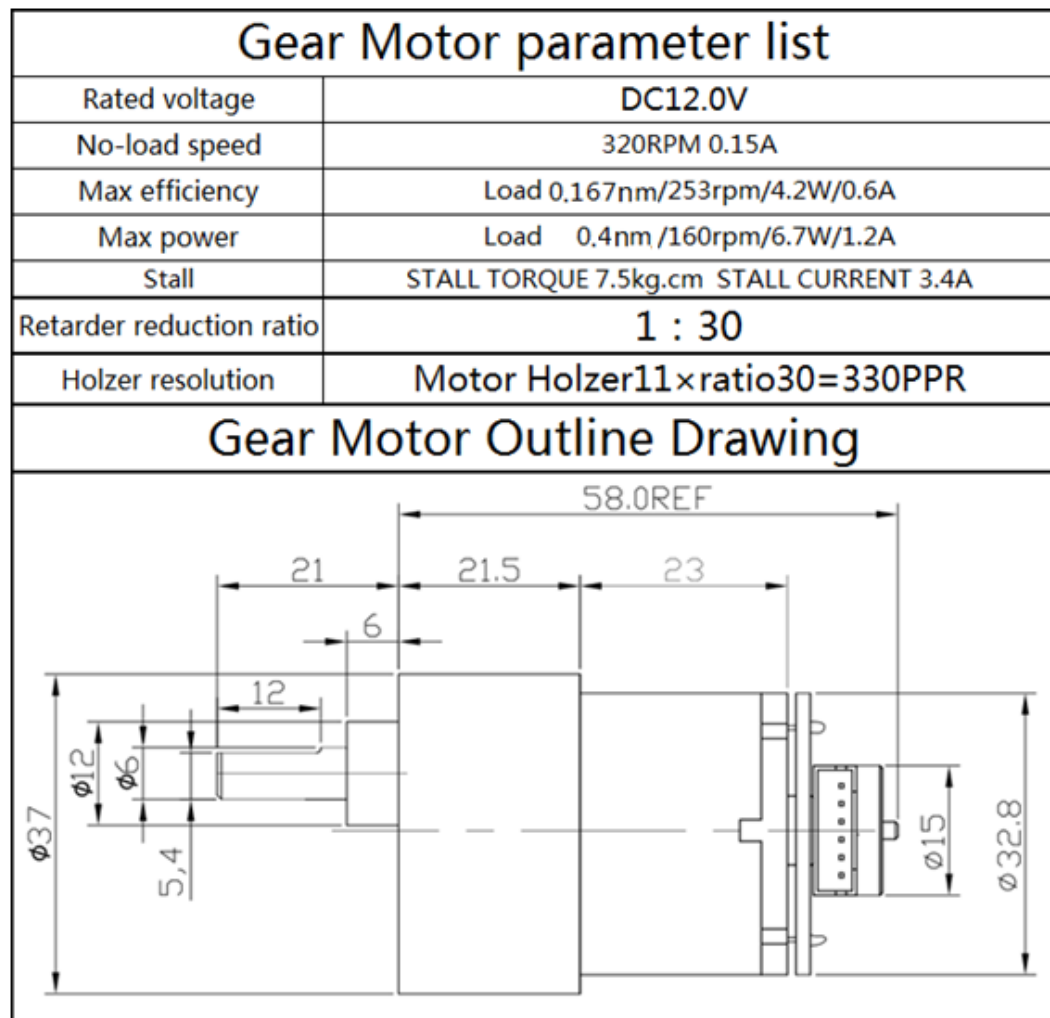
Modularcircuits. 2011. H-bridge the basics. Luettu 22.4.2017. <http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/>

Savolainen, J. & Vaittinen, R. Sääätötekniikan perusteita. 2007. Helsinki: Hakapaino Oy.

Mäkelä, M., Soininen, L., Tuomola S. & Öistämö, J. 2012. Tekniikan kaavasto. Hämeenlinna: Tammertekniikka / Amk-kustannus Oy.

LIITTEET

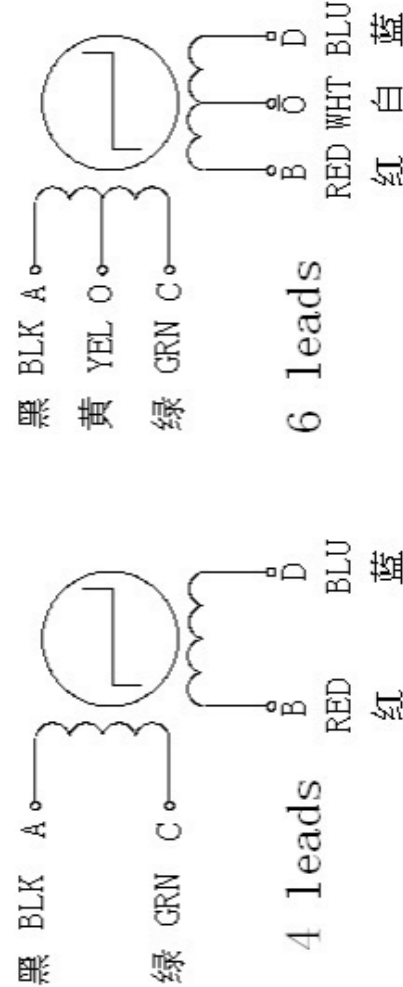
Liite 1. Moottorin datalehti



Liite 2. Askelmoottorin datalehti.

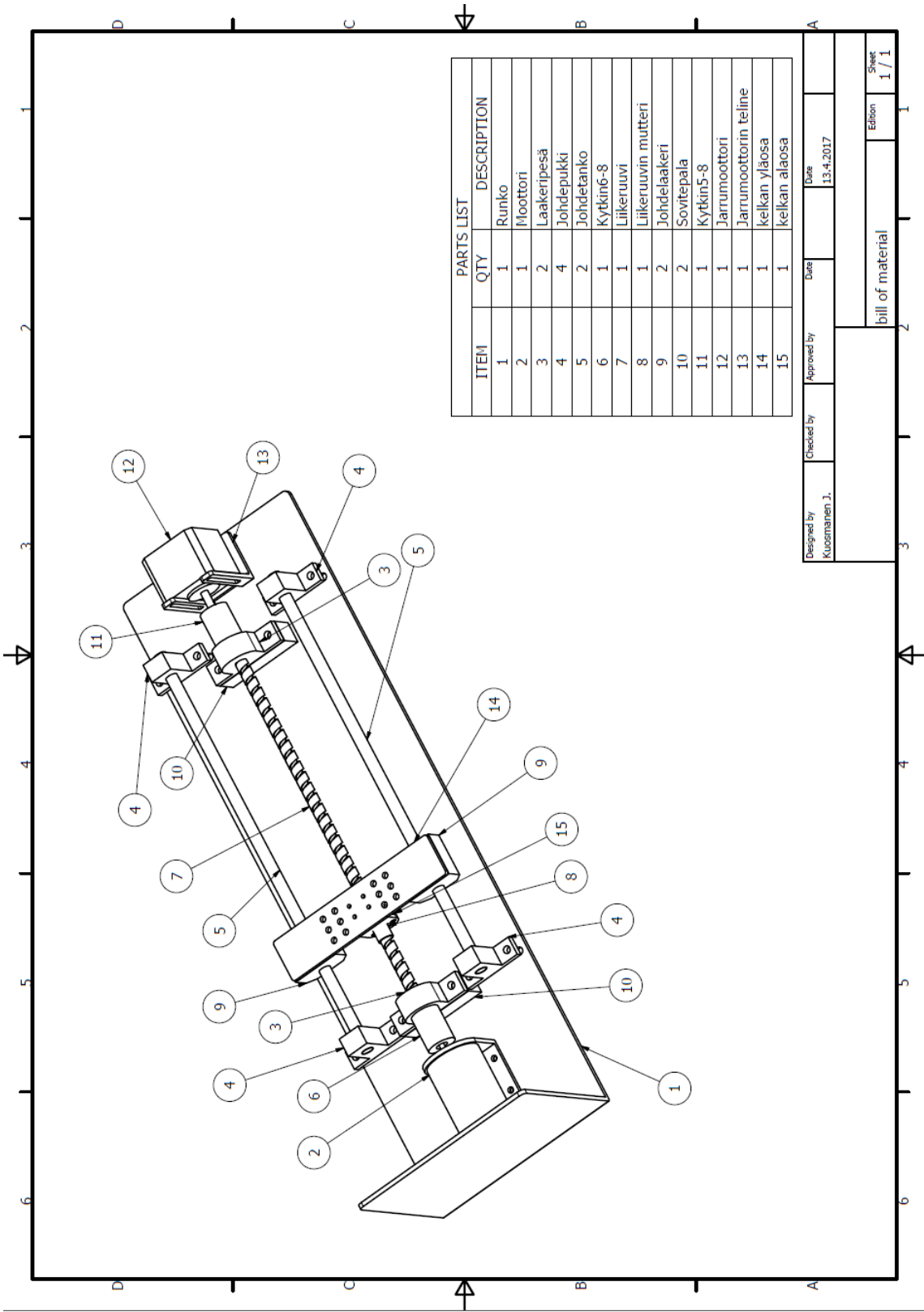
42BYGHW208 Stepper Motor Datasheet

Wiring diagram (接线图):

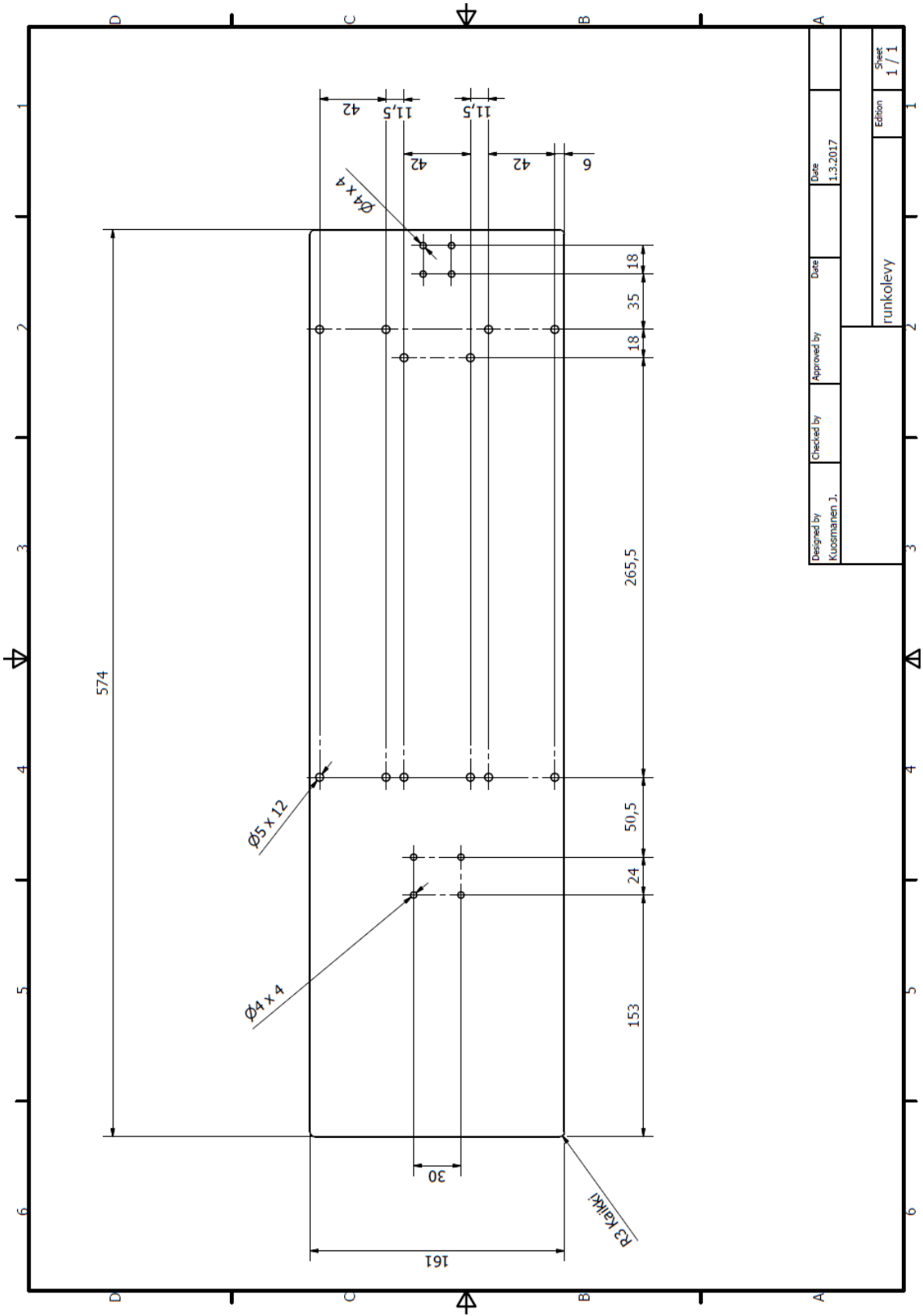


Model	Step Angle (°)	Motor Length L(mm)	Rate Voltage (V)	Rate Current (A)	Phase Resistance (Ω)	Phase Inductance (mH)	Holding Torque (g.cm)	Lead Wire (NO.)	Rotor Inertia (g.cm ²)	Detent Torque (g.cm)	Motor Weight (kg)
42BYGHW208	1.8	34	12	0.4	30	37	2800	4	34	200	0.2

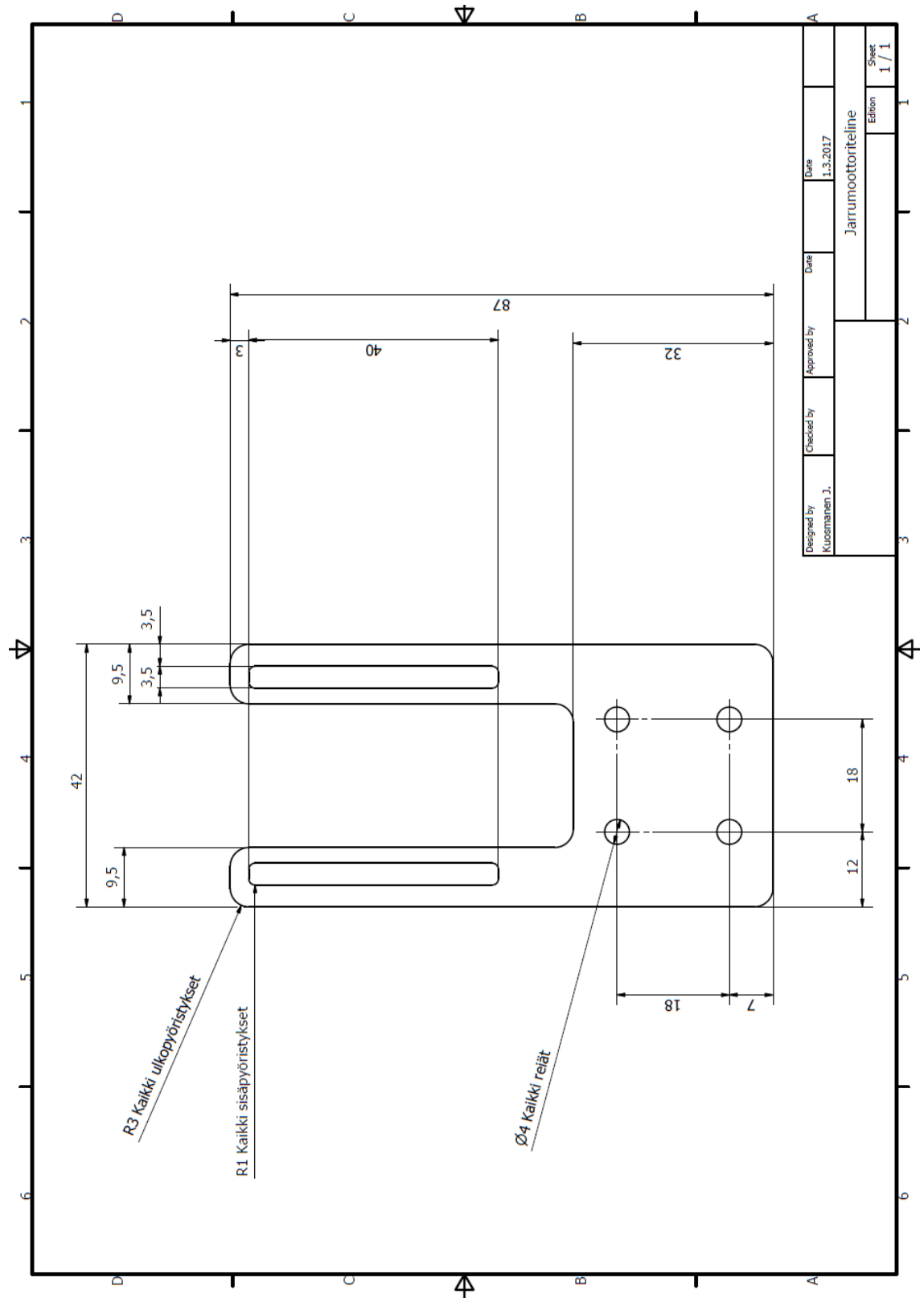
Liite 3. Kokoonpanokuva, jossa osat erikseen merkattuina



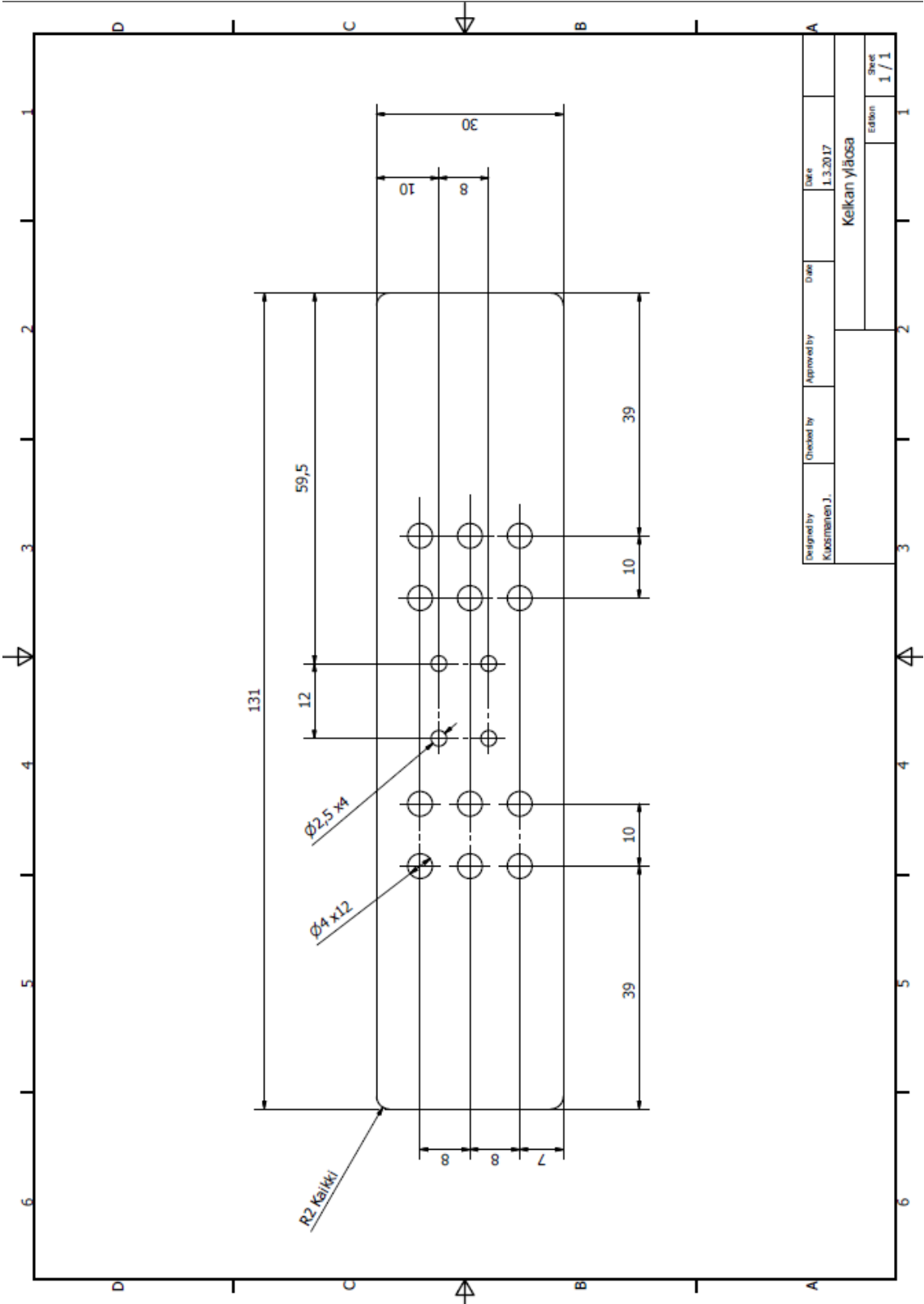
Liite 4. Runkolevyn mitoitettu 2D-malli



Liite 5. Jarrumoottoritelineen mitoitettu 2D-malli



Liite 6. Kelkan yläosan mitoitettu 2D-malli



Liite 9. Testiohjelma

```
/*  
Testiohjelma paikka-ajoon, ohjelma ajaa ensin kotiasemaan ja sen  
jälkeen  
haluttuun paikkaan määritellyllä toleranssilla  
  
HUOM! pinnit on määritetty Arduino UNO:n pinnikonfiguraation pe-  
rusteella  
eri arduinoissa käytetään eri pinnejä keskeytystoiminnoille ja  
pwm:lle  
*/  
  
// Enkooderin pinnit  
const int a_pulssi = 2;  
const int b_pulssi = 3;  
  
//H-sillan kontrollipinnit  
const int hp_1 = 7;  
const int hp_2 = 8;  
  
//PWM-pinni  
const int nopeussaato = 5;  
  
//kotiasemaan ajon rajakytkimen pinni  
const int rajakytkin = 12;  
  
//muuttuja johon tallennetaan enkooderin perusteella saatu paikka  
volatile long paikka = 0;  
//muuttuja johon tallennetaan mihin paikkaan halutaan ajaa, etäi-  
syys kotiasemasta millimetreinä
```

```
const int matka = 150;

//millimetreistä pulsseiksi muutettu haluttu paikkatieto
const int tavoitepaikka = matka * 330;

//paikkaan ajamisen tarkkuus millimetreinä
const int toleranssi_mm = 2;

//tarkkuus muutettuna pulsseiksi, kun nousu on 8mm/kierros
const int toleranssi = ((toleranssi_mm / 8) * 330)/2;

//kotiasemaan ajon ehto, jolla varmistetaan että aliohjelma suoritetaan vain kerran
int k_ehto = 0;

void setup() {

    pinMode(a_pulssi, INPUT);
    pinMode(b_pulssi, INPUT);

    pinMode(hp_1, OUTPUT);
    pinMode(hp_2, OUTPUT);

    pinMode(rajakytkin, INPUT);

    //keskeytystoiminnon käyttöönotto, keskeytys tapahtuu aina kun
    a_pulssin tila muuttuu
    //ja siirrytään enkooderi aliohjelmaan
    attachInterrupt(0, enkooderi, CHANGE);

}
```

```

void loop() {

    //kotiasemaan ajo ensimmäisellä ohjelmakierrolla
    if (k_ehto == 0){
        kotiasema;
    }

    //ajaminen haluttuun paikkaan

    if ( (tavoitepaikka - toleranssi) > paikka){

        ajo_eteenpain(150);
    }

    if ( (tavoitepaikka + toleranssi) < paikka){

        ajo_taksepain(150);
    }

    if ( (tavoitepaikka - toleranssi) < paikka < (tavoitepaikka +
    toleranssi) ){

        moottorin_vapautus;
    }

}

void kotiasema(){

    if (digitalRead(rajakytkin) == HIGH){
        //jos nc-kytketty rajakytkin ei ole vaikutettuna, ajetaan
        taaksepäin
        ajo_taksepain(150);
    }
}

```

```

while(digitalRead(rajakytkin) == HIGH){
    //ajetaan tyhjää while-silmukkaa kunnes rajakytkimen sig-
naali muuttu low-tilaan
}
//pysäytetään moottori
moottorin_vapautus;

//nollataan paikkatieto
k_ehto = 1;
paikka = 0;

}

}

```

```

//aliohjelma jolla ajetaan moottoria eteenpäin
void ajo_eteenpain(int nopeus) {
    //asetetaan pwm-pinnin lähtöarvo aliohjelmaan tuodulla säätöar-
volla
    analogWrite(nopeussaato, nopeus);
    //asetetaan h-siltaa ohjaavat pinnin haluttuun arvoon
    digitalWrite(hp_1, HIGH);
    digitalWrite(hp_2, LOW);
}

```

```

//aliohjelma jolla ajetaan moottoriataaksepäin
void ajo_taksepain(int nopeus) {
    analogWrite(nopeussaato, nopeus);
    digitalWrite(hp_1, LOW);
    digitalWrite(hp_2, HIGH);
}

```

```

//aliohjelma jolla asetetaan moottori vapaaseen tilaan
void moottorin_vapautus() {

```



```

//asetetaan pwm-pinnin arvo nolllaksi
  analogWrite(nopeussaato, 0);
  digitalWrite(hp_1, LOW);
  digitalWrite(hp_2, LOW);
}

//keskeytyksestä ajettava aliohjelma
void enkooderi() {

//tilanne jossa a_pulssi muuttu high-tilaan ja b_pulssi on low =
myötäpäivään oleva liike

    if (digitalRead(a_pulssi) == HIGH) {
        if (digitalRead(b_pulssi) == LOW) {
            paikka++;
        }
//tilanne jossa a_pulssi muuttu high-tilaan ja b_pulssi on high =
vastapäivään oleva liike
        else {
            paikka--;
        }
    } else {
//tilanne jossa a_pulssi muuttu low-tilaan ja b_pulssi on low =
vastapäivään oleva liike
        if (digitalRead(b_pulssi) == LOW) {
            paikka--;
        }
//tilanne jossa a_pulssi muuttu low-tilaan ja b_pulssi on high =
myötäpäivään oleva liike

        else {
            paikka++;
        }
    }
}

```

}
}